club commodore

Boletín informativo para los usuarios de microordenadores

VIC y CBM

editorial: hemos superado el milenario (pág. 1)

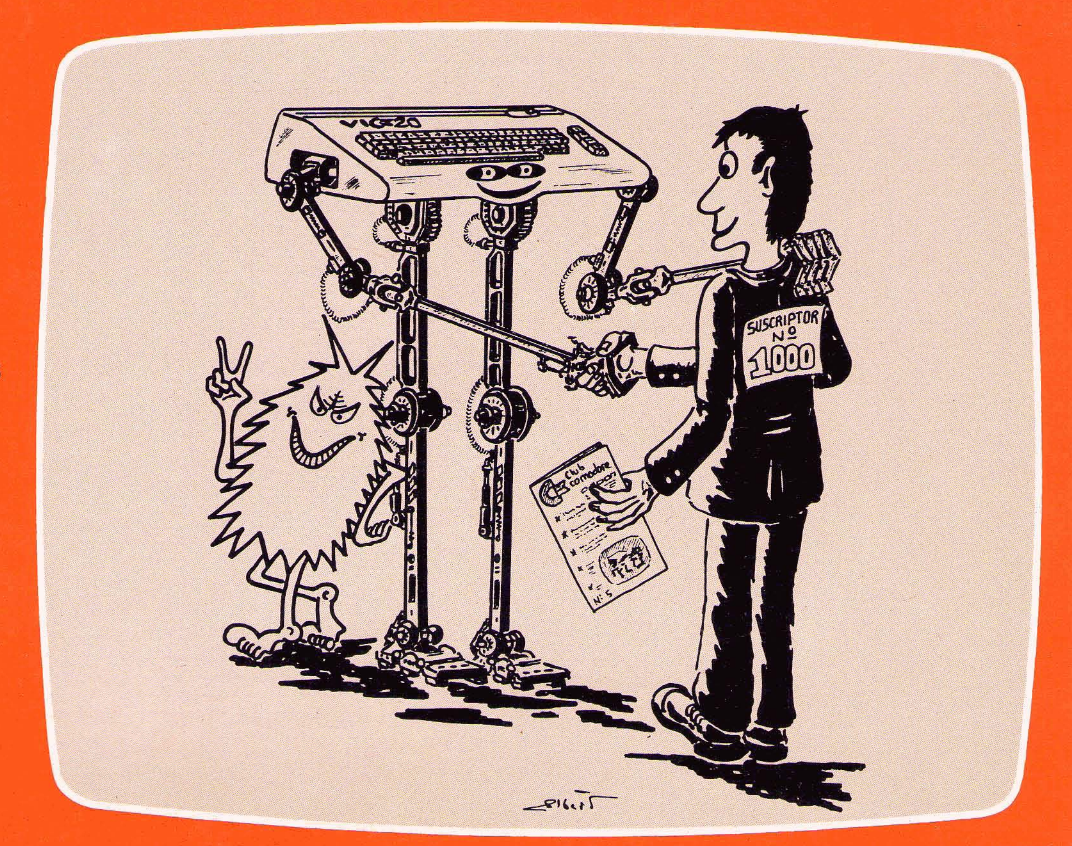
estructura de las cabeceras de variables y "arrays" (matrices) (pág. 1)

diseño de filtros para altavoces (pág. 3)

radioafición: RTTY para el VIC-20 (pág. 5)

grabación de datos en cassette (pág. 10)

software de base: técnicas de ordenación en memoria (pág. 14)

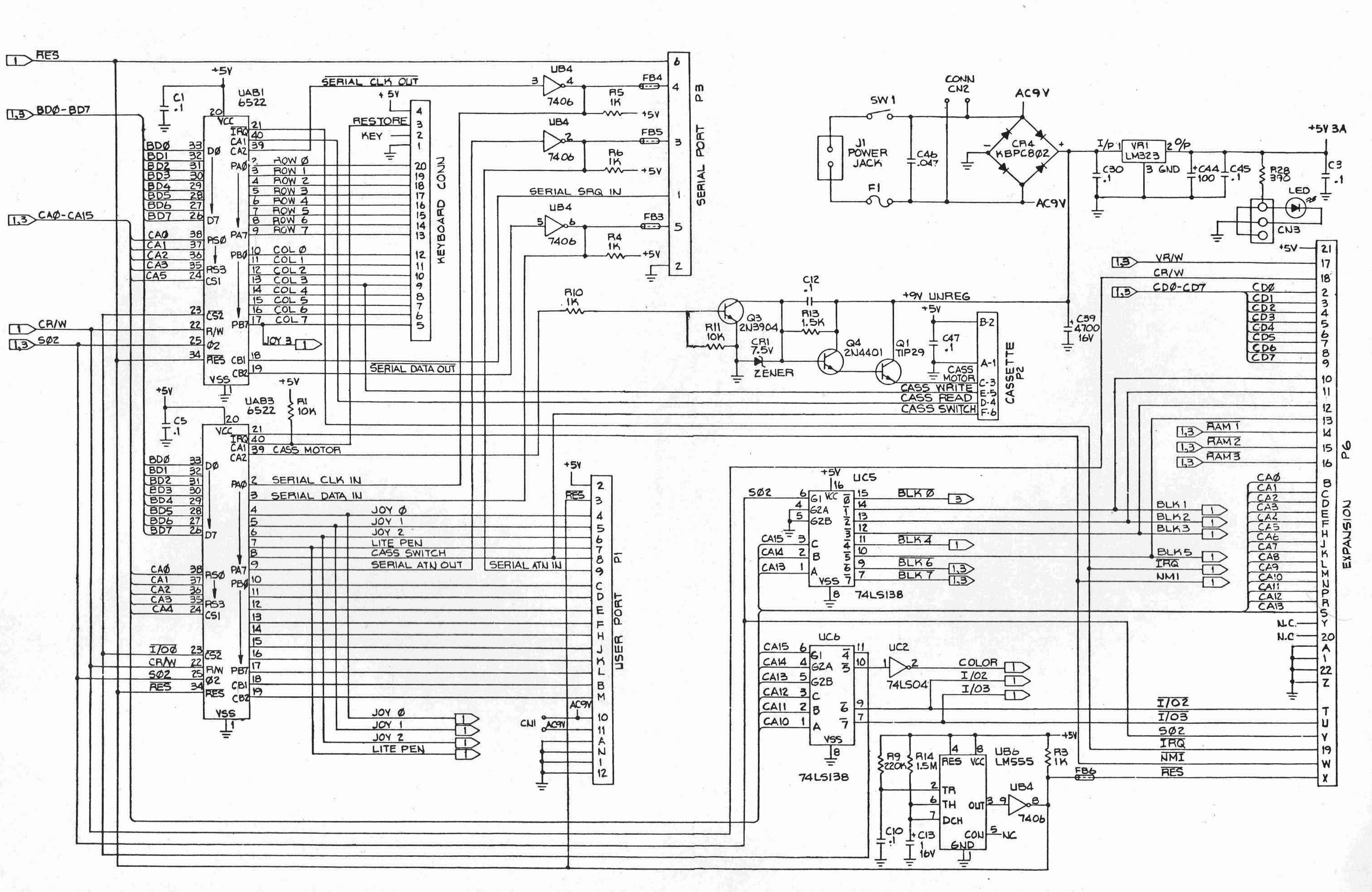




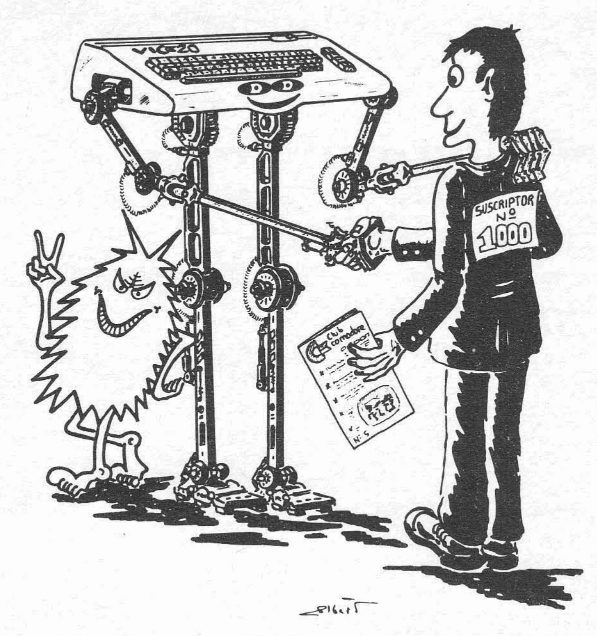
Nº 6

marzo 1983

ESQUEMA DEL VIC-20



EDITORIAL



¡hemos superado el milenario...!

...De suscripciones, claro. En estos momentos el número de lectores habituales y sistemáticos de "CLUB COMMODORE" supera la cifra de 1.000. Esto constituye todo un acontecimiento. El respaldo que esto significa para los que hacemos la Revista (no debe olvidarse que YA tenemos colaboraciones, aunque POCAS) es algo que queremos agradecer efusivamente.

Y ahora ¡basta de sentimentalismos! Desde un punto de vista práctico tenemos algo que significará una buena noticia para nuestros lectores: hemos desarrollado un programa que permite (suponemos) acabar definitivamente con los problemas causados por los listados y su lectura. Para más detalles sobre esta novedad véase el artículo: NUEVOS LISTADOS EN "CLUB COMMODORE".

Otra importante característica de este número 6 es la de que iniciamos la inserción de publicidad de los distribuidores y suministradores de software para los Ordenadores Personales COMMODORE. Aunque pueda parecer un tema frívolo éste de la publicidad tiene su importancia pues es el mejor método para conseguir el objetivo principal de nuestra Revista, que es ni más ni menos que el de suministrar al usuario un canal de información lo más completo posible sobre TODAS LAS ACTIVIDADES que se desarrollen en torno al equipo del que es poseedor.

Una nota final para los que nos remiten cartas con consultas de todo tipo: nos resulta totalmente imposible, dado el volumen de correspondencia que recibimos, contestar personalmente cada cuestión. Rogamos a nuestros corresponsales que tengan paciencia pues muchas de las consultas tienen contestación casi automática en la Revista al tratarse de temas que, o bien están en preparación, o van a aparecer en un futuro muy próximo. En resumen: disculpen las molestias y permanezcan atentos a la Revista. ■

VENTANA CBM

estructura de las cabeceras de variables y "arrays" (matrices)

por **JOAN CARLES SAMARANCH**

En este tercer y último artículo de la serie que he dedicado a mostrar la organización interna del BASIC de «Commodore», hablaré de los distintos tipos de variables existentes, bajo el punto de vista estricto de organización del sistema, sin entrar en planteamientos matemáticos binarios.

VARIABLES

Si directamente creamos la variable AB = 1.5, pasamos al monitor (SYS 1024) y visualizamos:

```
.M 0403 0403
.: 0403 41 42 81 40 00 00 00 AA
```

Observando la figura 1.b para variables en coma flotante vemos que los dos primeros bytes corresponden a los caracteres en ASCII de la variable utilizada ($41 = A, $42 = B), el siguiente al exponente en formato por exceso 128 y los cuatro restantes a la mantisa (total 7 bytes).

Si escribimos — volviendo al BASIC — CLR:AB% = 15 (variable entera, fig. 1.a) y visualizamos (con el monitor):

```
.:0403 C1 C2 00 0F 00 00 00 AA
```

Los dos primeros bytes corresponden a los caracteres de la variable en curso más 128 ($80) y los dos siguientes al valor de la variable em-

(continua en la pag. siguiente)

estructura de las cabeceras...

(viene de la pág. anterior)

paquetada. Los tres bytes que quedan hasta llegar a siete no se utilizan en este caso.

Para las variables de «strings» (cadenas) (fig. 1.c) sólo se le suma 128 al segundo carácter de su nombre, p. ej.: CLR:AB$="CBM".

.: 0403 41 C2 03 FB 7F 00 00 AA

Tal y como se especifica en la figura 1.c, el tercer byte es la longitud de la cadena y los dos siguientes la dirección donde se encuentra ésta ($7FFB). Como recordaréis de meses anteriores, los «strings» pueden estar en la zona de programas o en la de «strings», como en este caso:

.M 7FFB 7FFB
.: 7FFB 43 42 4D 05 04 XX XX XX

Primero encontramos los tres bytes correspondientes al «string» y, a continuación, la dirección del tercer byte de la variable ($0405).

«ARRAYS»

Los «arrays», también conocidos por matrices, constan de una cabecera y de los distintos elementos de la matriz (fig. 2.a).

En la cabecera (fig. 2.b) los dos primeros bytes corresponden al nombre de la matriz. Siguiendo el mismo método de sumar 128 a uno y/o a otro según el tipo (ver apartado anterior), los dos siguientes bytes son la composición de la longitud hasta el inicio del siguiente «array» (necesario debido a que cada uno es de distinta longitud). El quinto byte corresponde a la dimensión dada al crearlo o utilizarlo por primera vez. Los bytes 6 y 7 corresponden al número de elementos de la primera dimensión definida con la instrucción DIM. Los bytes 8 y 9 indican el número de elementos de la segunda dimensión, en caso de existir. Si fuera de tres dimensiones, existirían los bytes 10 y 11, y así sucesivamente. Como ya es sabido, podemos utilizar matrices sin dimensionar siempre y cuando no superemos el índice 10, o sea 11 elementos (0-10).

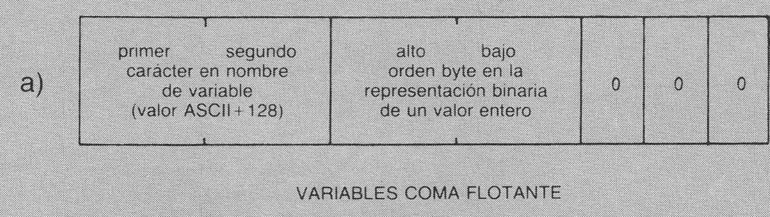
Pongamos por ejemplo: CLR:AB(0)=1.5, y consultemos al monitor la cabecera y el primer elemento:

.M 0403 0410
.: 0403 41 42 3E 00 01 00 0B 81
.: 040B 40 00 00 00 .. ..

(termina en la pág. siguiente)

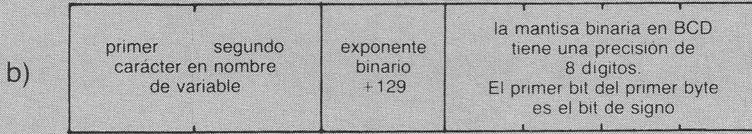
VARIABLES ENTERAS

a)



VARIABLES COMA FLOTANTE

b)



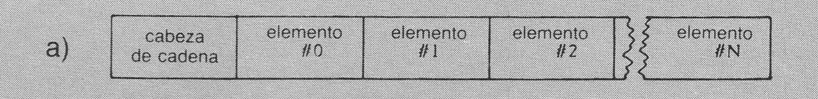
VARIABLES DE CADENA

c)

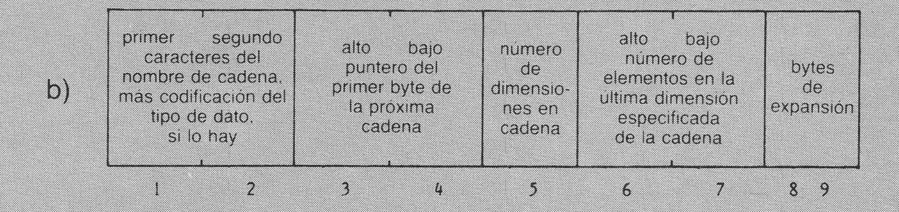
primer segundo carácter en nombre de variable, 128 añadido al valor ASCII del segundo carácter, solamente	número de caracteres	alto bajo orden de byte de dirección donde la cadena es almacenada	0	0

Fig. 1 - Estructura de las variables

a)

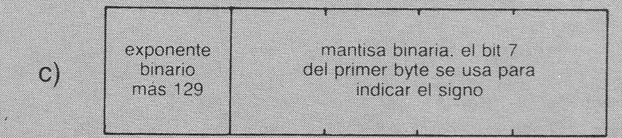


b)



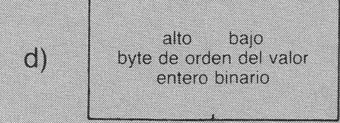
ELEMENTO CADENA DE COMA FLOTANTE

c)



ELEMENTO CADENA ENTERA

d)



ELEMENTO CADENA CARÁCTER

e)

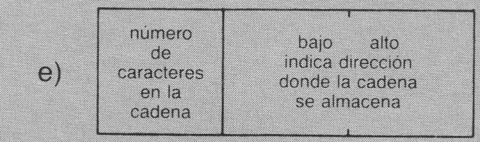


Fig. 2 - Estructura de los "arrays"

ELECTRÓNICA

diseño de filtros para altavoces

por **R. PARDO**

Este programa está concebido para calcular filtros para altavoces de dos y tres vías y de 6 ó 12 dB/octava indistintamente.

Los datos que pide el programa son:

RL (ohmios): valor de la impedancia del altavoz.

Ft (Hz): valor de la frecuencia de corte del filtro (dos vías).

FT. GRAVES (Hz): valor de la frecuencia de corte más grave (tres vías).

FT. AGUDOS (Hz): valor de la frecuencia de corte más aguda (tres vías).

Todos los resultados vienen dados en Henrios y microfaradios.

6 dB/oct
L1 C1 Gr Ag
L1 L2 C1 C2 Gr Me Ag
12 dB/oct
L1 C2 C1 L2
L1 C2 C4 L3 L2 L4 C3 C1

FUNCIONAMIENTO

Las líneas 30-90 le piden que elija entre un tipo de filtro u otro. Una vez elegido, el programa le preguntará el número de vías del diseño (líneas 100-150/360-420). Hecha la elección, el programa calcula los valores correspondientes de L y C y presenta los valores (líneas 200-230/300-350/470-520/610-710). Sigue el programa, y éste nos pregunta si deseamos iniciar un nuevo diseño, terminar el diseño o repetir los resultados anteriores, en cuyo caso saldrán en pantalla los resultados que antes habíamos obtenido. Si elige «NUEVO PROYECTO», el programa empieza de nuevo; y si no se desea seguir, el programa termina.

(termina en la pág. siguiente)

estructura de las cabeceras...

(viene de la pág. anterior)

Los dos primeros caracteres corresponden al nombre de la variable (prueba con matrices enteras y «strings» para ver la diferencia). El tercero y cuarto byte dan $003E = 62 bytes de longitud desglosados de la siguiente manera: 5 bytes (cabecera) + 5 bytes /elem. (por ser en coma flotante, figura 2.c) × 11 elementos. El sexto y séptimo expresan el número de elementos ($000B = 11), de la primera y única dimensión en este caso (expresado en el quinto byte = 1). Los bytes del 8 al 12 corresponden al valor asignado según el formato de la figura 2.c.

Los ejemplos correspondientes a los otros tipos de «arrays», matrices de más de una dimensión, distintos números de elementos, etc., os lo dejo que lo comprobéis vosotros con los datos dados y los formatos de la fig. 2.

ADAPTACIÓN AL VIC-20

El Sistema Operativo del VIC-20 funciona exactamente con la misma filosofía comentada en esta serie de artículos y sólo varían algunas direcciones y posiciones concretas de memoria.

He aquí las posiciones utilizadas en los ejemplos:

	PET	VIC
Principio BASIC	40-41	43-44
Principio variables	42-43	45-46
Principio «arrays»	44-45	47-48
Final «arrays»	46-47	49-50
Almacenamiento de «strings»	48-49	51-52
Final de memoria	52-53	55-56
Monitor	SYS1024	SYS24576 (con el cartucho Monitor de Lenguaje Maquina)

(viene de pág. anterior)

FÓRMULAS UTILIZADAS

DOS VÍAS

```
L1=RL/(2*[pi]*FT)

C1=1/(2*[pi]*FT*RL)

L1=(RL*SQR(2))/(2*[pi]*FT)

C1=1/(2*[pi]*FT*RL*SQR(2))

L2=RL/(2*[pi]*FT2)

C2=1/(2*[pi]*FT2*RL)
```

TRES VÍAS

```
L1=RL/(2*[pi]*FT1)

C1=1/(2*[pi]*FT1*RL)

L1=(RL*SQR(2))/(2*[pi]*FT1)

C1=1/(2*[pi]*FT1*RL*SQR(2))

L3=(RL*SQR(2))/(2*[pi]*FT2)

C3=1/(2*[pi]*FT2*Rl*SQR(2))
```

NOTA.—En estas fórmulas [pi] equivale al valor de la variable π.

MUESTRAS DE LOS RESULTADOS

6 dB/Octv. 2 Vías

RL = 8 Ω
Ft = 4500 Hz

L1 = .282942121 μH
C1 = 4.42097064 μF

3 Vías

RL = 8 Ω
Ft graves = 2000 Hz
Ft agudos = 7000 Hz

L1 = .636619773 μH
L2 = .181891364 μH
C1 = 9.94718395 μF
C2 = 2.84205256 μF

12 dB/Oct. 2 Vías

RL = 8 Ω
Ft = 4500 Hz

L1 = .400140585 μH
L2 = .400140585 μH
C1 = 3.12609832 μF
C2 = 3.12609832 μF

12 dB/Oct. 3 Vías

RL = 8 Ω
Ft graves = 2000 Hz
Ft agudos = 7000 Hz

L1 = .900316317 μH
L2 = .900316317 μH
L3 = .257233233 μH
L4 = .257233233 μH
C1 = 7.03372123 μF
C2 = 7.03372123 μF
C3 = 2.00963464 μF
C4 = 2.00963464 μF

■

```
READY.

0 POKE36869,PEEK(36869)+2
10 REM FILTRO ALTAVOCES
20 REM R.PARDO/CLUB COMMODORE/16.10.1982
30 PRINT"[ CLR ][RVSON][SHIFF][SHIFI][SHIFL]
[SHIFT][SHIFR][SHIFO][SHIFS] [SHIFP][SHIFA]
[SHIFR][SHIFA] [SHIFA][SHIFL][SHIFT][SHIFA]
[SHIFV][SHIFO][SHIFC][SHIFE][SHIFS][RVSOF]"
40 PRINT:PRINT"1-[SHIFF]ILTRO 6D[SHIFB]/OCT
AVA"
50 PRINT:PRINT"2-[SHIFF]ILTRO 12D[SHIFB]/OC
TAVA"
60 PRINT:PRINT"[SHIFP]ULSE 1 O 2"
70 GETA$:IFA$="1"THEN100
80 IFA$="2"THEN370
90 GOTO70
100 PRINT"[ CLR ]1-[SHIFD]OS VIAS"
110 PRINT:PRINT"2-[SHIFT]RES VIAS"
120 PRINT:PRINT"[SHIFP]ULSE 1 O 2"
130 GETB$:IFB$="1"THEN160
140 IFB$="2"THEN250
150 GOTO130
160 PRINT"[ CLR ]":INPUT"[SHIFR][SHIFL](OHM
S)";RL
170 PRINT:INPUT"[SHIFF]T([SHIFH]Z)";FT
190 L1=(RL/(2*[COMM↑]*FT))*1000
190 C1=1E+6/(2*[COMM↑]*FT*RL)
200 PRINT"[ CLR ][RVSON][SHIFC][SHIFA][SHIFL]
[SHIFC][SHIFU][SHIFL][SHIFO] [SHIFF][SHIFI]
[SHIFL][SHIFT][SHIFR][SHIFO][RVSOF]"
210 PRINT:PRINT"[SHIFL]1=";L1;"U[SHIFH]"
220 PRINT:PRINT"[SHIFC]1=";C1;"U[SHIFF]"
230 GOSUB800
240 GOSUB720:GOTO180
250 PRINT"[ CLR ]":INPUT"[SHIFR][SHIFL](OHM
S)";RL
260 PRINT:INPUT"[SHIFF]T GRAVES([SHIFH]Z)";
F1
270 PRINT:INPUT"[SHIFF]T AGUDOS([SHIFH]Z)";
F2
280 L1=(RL/(2*[COMM↑]*F1))*1E+3:L2=(RL/(2*[COMM↑]
*F2))*1E+3
290 C1=1E+6/(2*[COMM↑]*F1*RL):C2=1E+6/(2*[COMM↑]
*F2*RL)
300 PRINT"[ CLR ][RVSON][SHIFC][SHIFA][SHIFL]
[SHIFC][SHIFU][SHIFL][SHIFO] [SHIFF][SHIFI]
[SHIFL][SHIFT][SHIFR][SHIFO][RVSOF]"
310 PRINT:PRINT"[SHIFL]1=";L1;"U[SHIFH]"
320 PRINT:PRINT"[SHIFL]2=";L2;"U[SHIFH]"
330 PRINT:PRINT"[SHIFC]1=";C1;"U[SHIFF]"
340 PRINT:PRINT"[SHIFC]2=";C2;"U[SHIFF]"
350 GOSUB800
360 GOSUB720:GOTO300
370 PRINT"[ CLR ]1-[SHIFD]OS VIAS"
380 PRINT:PRINT"2-[SHIFT]RES VIAS"
390 PRINT:PRINT"[SHIFP]ULSE 1 O 2"
400 GETC$:IFC$="1"THEN430
410 IFC$="2"THEN540
420 GOTO400
430 PRINT"[ CLR ]":INPUT"[SHIFR][SHIFL](OHM
S)";RL
440 PRINT:INPUT"[SHIFF]T([SHIFH]Z)";FT
450 L1=((RL*SQR(2))/(2*[COMM↑]*FT))*1E+3:L2
=L1
460 C1=1E+6/(2*[COMM↑]*FT*RL*SQR(2)):C2=C1
470 PRINT"[ CLR ][RVSON][SHIFC][SHIFA][SHIFL]
[SHIFC][SHIFU][SHIFL][SHIFO] [SHIFF][SHIFI]
[SHIFL][SHIFT][SHIFR][SHIFO][RVSOF]"
480 PRINT:PRINT"[SHIFL]1=";L1;"U[SHIFH]"
490 PRINT:PRINT"[SHIFL]2=";L2;"U[SHIFH]"
500 PRINT:PRINT"[SHIFC]1=";C1;"U[SHIFF]"
510 PRINT:PRINT"[SHIFC]2=";C2;"U[SHIFF]"
520 GOSUB800
530 GOSUB720:GOTO470
540 PRINT"[ CLR ]":INPUT"[SHIFR][SHIFL](OHM
S)";RL
550 PRINT:INPUT"[SHIFF]T GRAVES([SHIFH]Z)";
F1
560 PRINT:INPUT"[SHIFF]T AGUDOS([SHIFH]Z)";
F2
570 L1=((RL*SQR(2))/(2*[COMM↑]*F1))*1E+3:L2
=L1
580 L3=((RL*SQR(2))/(2*[COMM↑]*F2))*1E+3:L4
=L3
590 C1=1E+6/(2*[COMM↑]*F1*RL*SQR(2)):C2=C1
600 C3=1E+6/(2*[COMM↑]*F2*RL*SQR(2)):C4=C3
610 PRINT"[ CLR ][RVSON][SHIFC][SHIFA][SHIFL]
[SHIFC][SHIFU][SHIFL][SHIFO] [SHIFF][SHIFI]
[SHIFL][SHIFT][SHIFR][SHIFO][RVSOF]"
620 PRINT:PRINT"[SHIFL]1=";L1;"U[SHIFH]"
630 PRINT:PRINT"[SHIFL]2=";L2;"U[SHIFH]"
640 PRINT:PRINT"[SHIFL]3=";L3;"U[SHIFH]"
650 PRINT:PRINT"[SHIFL]4=";L4;"U[SHIFH]"
660 GOSUB800
670 PRINT:PRINT"[SHIFC]1=";C1;"U[SHIFF]"
680 PRINT:PRINT"[SHIFC]2=";C2;"U[SHIFF]"
590 PRINT:PRINT"[SHIFC]3=";C3;"U[SHIFF]"
700 PRINT:PRINT"[SHIFC]4=";C4;"U[SHIFF]"
710 GOSUB800:GOSUB720:GOTO610
720 PRINT"[ CLR ]1-[SHIFR]EPETICION DATOS"
730 PRINT:PRINT"2-[SHIFN]UEVO PROYECTO"
740 PRINT:PRINT"3-[SHIFF]IN DE PROYECTO"
750 PRINT:PRINT"[SHIFP]ULSE 1,2 O 3"
760 GETL$:IFL$=""THEN760
770 K=VAL(L$):ONKGOTO780,30,790
780 RETURN
790 POKE36869,PEEK(36869)-2:END
800 PRINT:PRINT"[SHIFP]ULSE UNA TECLA"
810 GETK$:IFK$=""THEN810
820 RETURN
READY.
```

RADIOAFICIÓN

RTTY para el VIC-20



por **EA3CIW**

Esta colaboración, que nos manda EA3CIW, consiste en un pequeño programa de recepción de señales de RTTY aprovechando todo el protocolo RS232 que incorpora el VIC-20.

El sistema presentado está basado en los libros THE VIC REVEALED y PROGRAMMERS REFERENCE GUIDE, en los cuales hay suficiente información para ver el manejo del canal RS 232 desde BASIC, así como desde código máquina, que es el número 2; y en distintas revistas y publicaciones para obtener el alfabeto Baudot y sus conversiones.

En la línea 100 del programa de la figura 1 se abre el canal RS232 para 5 bits, 1 bit de stop, 50 bauds y «half duplex». Podemos programar directamente en el VIC cualquier velocidad de transmisión de las siguientes: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800 y 2400 bauds, y utilizar desde 5 hasta 8 bits. Esto quiere decir que por el mismo método, sólo cambiando el programa, podremos recibir emisiones en ASCII a 110 bauds.

De todas maneras, para poder recibir correctamente las estaciones de radioaficionados que transmiten a 45.45 bauds, se incorpora la línea 120 que modifica la base de tiempos. En el caso de querer escuchar estaciones profesionales de información que transmitan a 50 bauds deberemos eliminar esta línea.

Una de las que mejor se reciben es «The Associated Press London» que transmite a 50 bauds en la frecuencia de 6.985,3 KHz y en «reverse». Esto quiere decir que escuchada en LSB (banda lateral inferior), como es costumbre, el tono agudo corresponde al espacio. En la mayoría de demoduladores se incorpora esta opción que consiste en un simple inversor.

El resto del programa consta de la preparación de la tabla de conversión de los caracteres del alfabeto Baudot a los normales del VIC-20 (líneas 500-530) y de la parte editora de los textos.

Como es frecuente que haya fading y/o ruidos se permite que, pulsando la tecla de función F2, se pueda cambiar de números a letras y viceversa durante la recepción.

En la figura 2 se muestra el esquema de bloques del modulador-demodulador que ha preparado EA3CIW (Box 482, Sabadell), que incorpora: «reverse» y/o normal, PTT («push to talk»), transmisión directa en FSK, si el transceptor lo incorpora, y modulador para hacerlo vía audio o en el caso de trabajar RTTY en FM. Para el trabajo en CW utiliza el PTT para manejar la emisora y el filtro del demodulador para la recepción. ■



```
RTTY-RX 2.2

10 REM *** RTTY-RX 2.2 **
20 REM *** POR EA3CIW ***
100 OPEN2,2,2,CHR$(97)+CHR$(16):GET#2,A$
:POKE36876,241
110 DIM M%(31,1)
120 POKE665,122:POKE666,95:REM*** 45.45
BAUDS ***
500 DATA0,0,69,51,17,17,65,45,32,32,83,3
9,73,56,85,55,13,13,68,42,82,52
510 DATA74,7,78,44,70,35,67,58,75,40,84,
53,90,43,76,41,87,50
520 DATA72,36,89,54,80,48,81,49,79,57,66
,63,71,37,0,0,77,46,88,47,86,61,0,0
530 FORI=0TO31:READM%(I,0),M%(I,1):NEXT:
D=0
1000 GET#2,C$:C$=C$+CHR$(0)
1010 C=ASC(C$)
1020 IFC=27THEND=1:GOTO1000
1025 IFC=31THEND=0:GOTO1000
1030 GETA$:IFA$<>CHR$(134)THEN1040
1035 IFDTHEND=0:GOTO1040
1037 IFD=0THEND=1
1040 REM
2000 E=M%(C,D):IFE=7THEN2100
2020 IFE=17ANDF=13THEN1000
2050 PRINTCHR$(E);:F=E:GOTO1000
2100 POKE36878,15:FORI=1TO100:NEXT:POKE3
6878,0:GOTO1000

READY.
```

Fig. 1

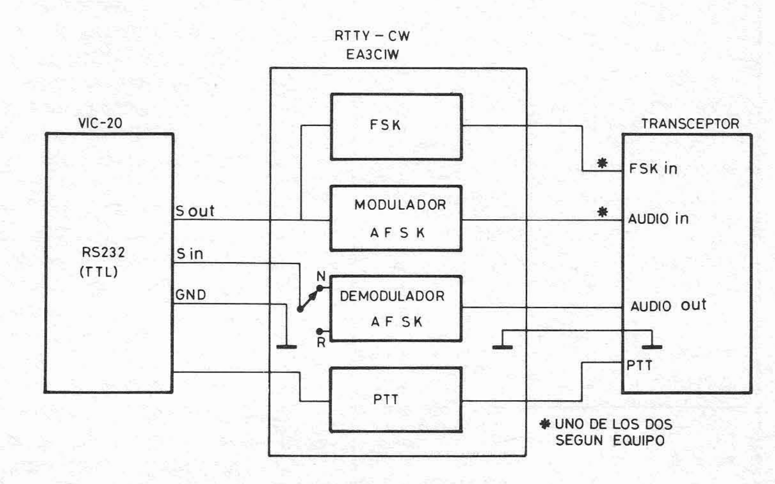


Fig. 2

los nuevos listados de "CLUB COMMODORE"

por **PERE MASATS**

En una reunión del nuevo Club de Usuarios de Barcelona (por cierto fue en la primera) al preguntar la opinión de los presentes sobre nuestra Revista y después de montañas de elogios a CLUB COMMODORE (¡EJEM!) la crítica más enérgica (y prácticamente la única) fue dedicada a los listados de nuestra Revista y concretamente a la dificultad de leerlos. En consecuencia y luego de discutir largamente el asunto con el Equipo Redactor Habitual se ha llegado a dos importantes conclusiones, a saber:

PRIMERA. — Hay que hacer algo respecto al tema de los listados y

SEGUNDA y no menos importante: LOS CLUBS DE USUARIOS SON MUY ÚTILES PARA OBLIGARNOS A AFRONTAR LAS DURAS REALIDADES DE LA VIDA MICROINFORMÁTICA.

En lo que respecta a la primera, a continuación se presenta nuestra solución; y en cuanto a la segunda... Los **USUARIOS TIENEN LA PALABRA.**

La solución en principio debe reunir ciertas características:

A) El listado debe hacerse a través de un ordenador, no sólo por razones de presentación sino porque es la manera de evitar errores en los listados.

```
LIST DE LIST, VERSION ANTIGUA

READY.

 0 REM PROGRAMA PARA HACER LISTADOS (CBM 8023)
 10 H$="":X=0
 20 PRINT"▯PROGRAMA PARA LISTAR PROGRAMAS"
 30 INPUT"NOMBRE";S$
 40 INPUT"CARACTERES";B
 50 OPEN1,9,2,"0:"+S$+",S,R":OPEN4,4:PRINT#4,CHR$(0)
 51 OPEN13,4,13:PRINT#13:CLOSE13
 52 OPEN15,4,15:PRINT#15:CLOSE15
 60 PRINT#4,"PROGRAMA "+S$
 70 PRINT#4,CHR$(1)
 100 IFST=64THENCLOSE1:PRINT#4:CLOSE4:RUN
 110 GET#1,A$
 120 GOSUB170
 130 H$=H$+A$
 140 IF(LEN(H$)>=B)OR(A$=CHR$(13))THENPRINT#4,H$:PRINTH$:H$=""
 150 REMIFLEN(H$)=BTHENPRINT#4,H$:PRINTH$:H$=""
 160 GOTO100
 170 IFA$="▯"THENA$="[CRSRD]":RETURN
 180 IFA$="▯"THENA$="[CRSRU]":RETURN
 190 IFA$="▯"THENA$="[CRSRR]":RETURN
 200 IFA$="▯"THENA$="[CRSRL]":RETURN
 210 IFA$="▯"THENA$="[HOME ]":RETURN
 220 IFA$="▯"THENA$="[ CLR ]":RETURN
 230 IFA$="▯"THENA$="[ BLK ]":RETURN
 240 IFA$="▯"THENA$="[ WHT ]":RETURN
 250 IFA$="▯"THENA$="[ RED ]":RETURN
 260 IFA$="◣"THENA$="[ CYN ]":RETURN
 270 IFA$="▯"THENA$="[ PUR ]":RETURN
 280 IFA$="▯"THENA$="[ GRN ]":RETURN
 290 IFA$="▯"THENA$="[ BLU ]":RETURN
 300 IFA$="▯"THENA$="[ YEL ]":RETURN
 310 IFA$="▯"THENA$="[RVSON]":RETURN
 320 IFA$="▯"THENA$="[RVSOF]":RETURN
 330 IFA$="▯"THENA$="[INST ]":RETURN
 340 IFA$="▯"THENA$="[CTRL←]":RETURN
 350 REMIFA$=":"THENA$=":"+CHR$(13)
 360 IFA$="▒"THENA$="[COMM+]":RETURN
 370 IFA$="┼"THENA$="[SHIF+]":RETURN
 380 IFA$="▒"THENA$="[COMM-]":RETURN
 390 IFA$="│"THENA$="[SHIF-]":RETURN
 400 IFA$="▒"THENA$="[COMM£]":RETURN
 410 IFA$="◤"THENA$="[SHIF£]":RETURN
 420 IFA$="├"THENA$="[COMMQ]":RETURN
 430 IFA$="●"THENA$="[SHIFQ]":RETURN
 440 IFA$="┤"THENA$="[COMMW]":RETURN
 450 IFA$="○"THENA$="[SHIFW]":RETURN
 460 IFA$="┴"THENA$="[COMME]":RETURN
 470 IFA$="‾"THENA$="[SHIFE]":RETURN
 480 IFA$="┬"THENA$="[COMMR]":RETURN
 490 IFA$="_"THENA$="[SHIFR]":RETURN
 500 IFA$="‾"THENA$="[COMMT]":RETURN
 510 IFA$="│"THENA$="[SHIFT]":RETURN
 520 IFA$="‾"THENA$="[COMMY]":RETURN
 530 IFA$=" │"THENA$="[SHIFY]":RETURN
 540 IFA$="▀"THENA$="[COMMU]":RETURN
 550 IFA$="╭"THENA$="[SHIFU]":RETURN
 560 IFA$="▄"THENA$="[COMMI]":RETURN
 570 IFA$="╮"THENA$="[SHIFI]":RETURN
 580 IFA$="▄"THENA$="[COMMO]":RETURN
 590 IFA$="┌"THENA$="[SHIFO]":RETURN
 600 IFA$="▁"THENA$="[COMMP]":RETURN
 610 IFA$="┐"THENA$="[SHIFP]":RETURN
 620 IFA$="_"THENA$="[COMM@]":RETURN
 630 IFA$="┘"THENA$="[SHIF@]":RETURN
 640 IFA$="◥"THENA$="[COMM*]":RETURN
 650 IFA$="─"THENA$="[SHIF*]":RETURN
 660 IFA$="π"THENA$="[COMM↑]":RETURN
 670 IFA$="π"THENA$="[SHIF↑]":RETURN
 680 IFA$=" ┌"THENA$="[COMMA]":RETURN
 690 IFA$="♠"THENA$="[SHIFA]":RETURN
 700 IFA$="┐"THENA$="[COMMS]":RETURN
 710 IFA$="♥"THENA$="[SHIFS]":RETURN
 720 IFA$=" ▗"THENA$="[COMMD]":RETURN
 730 IFA$="─"THENA$="[SHIFD]":RETURN
 740 IFA$="▖"THENA$="[COMMF]":RETURN
 750 IFA$="─"THENA$="[SHIFF]":RETURN
 760 IFA$="│"THENA$="[COMMG]":RETURN
 770 IFA$="│"THENA$="[SHIFG]":RETURN
 780 IFA$="│"THENA$="[COMMH]":RETURN
 790 IFA$=" │"THENA$="[SHIFH]":RETURN
 800 IFA$="▌"THENA$="[COMMJ]":RETURN
 810 IFA$=" ╰"THENA$="[SHIFJ]":RETURN
 820 IFA$="▌"THENA$="[COMMK]":RETURN
 830 IFA$="╯"THENA$="[SHIFK]":RETURN
 840 IFA$=" ▐"THENA$="[COMML]":RETURN
 850 IFA$="└"THENA$="[SHIFL]":RETURN
 860 IFA$=" └"THENA$="[COMMZ]":RETURN
 870 IFA$="♦"THENA$="[SHIFZ]":RETURN
 880 IFA$="┘"THENA$="[COMMX]":RETURN
 890 IFA$="♣"THENA$="[SHIFX]":RETURN
 900 IFA$=" ▝"THENA$="[COMMC]":RETURN
 910 IFA$="─"THENA$="[SHIFC]":RETURN
 920 IFA$="▘"THENA$="[COMMV]":RETURN
 930 IFA$="╳"THENA$="[SHIFV]":RETURN
 940 IFA$="▚"THENA$="[COMMB]":RETURN
 950 IFA$="│"THENA$="[SHIFB]":RETURN
 960 IFA$=" │"THENA$="[COMMN]":RETURN
 970 IFA$="╱"THENA$="[SHIFN]":RETURN
 980 IFA$=" │"THENA$="[COMMN]":RETURN
 990 IFA$=" │"THENA$="[COMMM]":RETURN
 1000 IFA$="╲"THENA$="[SHIFM]":RETURN
 1010 RETURN
READY.
```

B) Deben sustituirse todos los gráficos por algún tipo de mensaje que contenga la secuencia de teclas con las que se consigue el signo.

C) Por razones de impresión de la Revista debe limitarse el ancho del listado para poder «jugar» con los originales y obtener un fotolito lo más claro posible. (Este detalle es el menos evidente para el lector.)

D) El listado debe ser fácilmente «recortable» en diferentes secciones por personal sin conocimientos de programación BASIC —por cierto: un saludo para ellos— para adaptarlo a las necesidades de compaginación.

Atendiendo a estos parámetros se preparó el programa que se ilustra en la figura y que se da en las dos formas, antigua y actual.

El funcionamiento del programa es el siguiente: en el VIC (o en cualquier otro equipo COMMODORE dotado de una unidad de discos 4040) se carga o escribe el programa que se quiere listar, se corrigen todas las travesuras de BUG y, cuando estamos seguros de su funcionamiento, hacemos lo que sigue:

OPEN 1,8,2,"0:NOMBRE DEL PROGRAMA,S,W" y RETURN

Esto abre en la unidad de disco un fichero secuencial en escritura. Luego:

CMD1:LIST y RETURN

Así listamos el programa en forma alfanumérica en el disco, lo que nos servirá para su manejo posterior por nuestro programa. Cuando aparece READY tecleamos:

CLOSE1 y RETURN

Lo que nos cierra el fichero. Y ya tenemos nuestro programa listo para ser formateado y convenientemente listado. Obsérvese que el programa sigue siendo el mismo sin que realicemos sobre él ninguna copia o alteración manual.

Una vez tenemos el programa en forma de fichero secuencial en un disco de VIC (todo este procedimiento es fácilmente adaptable a otros equipos COMMODORE), pasamos a otro equipo dotado de una impresora sofisticada (la CBM 8023) que, entre otras cosas, permite la doble impresión, lo cual mejora los originales para los fotolitos. El equipo 8032 que también sirve con un programa de proceso de texto para la redacción de los artículos de CLUB COMMODORE, está conectado a una unidad de disco 4040 (compatible con el disco del VIC) además del suyo propio (un 8050); el 4040 es periférico 9 y el 8050 es el 8.

En la CPU 8032 se carga el programa que listamos en las dos versiones y su ejecución permite los nuevos listados. Veamos cómo trabaja:

La línea 10 inicializa las variables, la 30 pregunta el nombre del programa que tenemos en el disco y queremos listar.

La 40 nos pide el número de caracteres de ancho con que queremos que salga el listado. La 50 abre el fichero en la unidad de disco con el nombre que hemos dado; también inicia el funcionamiento de la impresora. El PRINT # 4,CHR$(0) es un resto del programa para funcionar con otro tipo de impresora y puede eliminarse. Las líneas 51-52 hacen que la impresora golpee dos veces cada carácter, lo que mejora la impresión (a costa de la velocidad, como es típico) y en caso de utilizar la del VIC pueden eliminarse. La línea 60 imprime al principio del listado el nombre del programa, lo que es de agradecer cuando se manejan unos cuantos originales al mismo tiempo. De la línea 70 diremos que es otro resto del trabajo con otra impresora.

La línea 100 busca el fin del fichero. Si lo encuentra, termina el listado, cierra el fichero y vuelve al principio del programa por si deseamos otro listado. La línea 110 toma un carácter del listado en el disco para procesarlo como la variable A$. En la línea 120 se salta a la subrutina 170 que es la que se encarga de sustituir el contenido de A$ por su equivalente alfanumérico si es un carácter gráfico. En la línea 130 se «suma» en alfabético el contenido de A$ al de H$ que es la que contiene el resto de la línea actual, en 140 si la longitud de H$ supera el número de caracteres que hemos asignado con la variable B o si el carácter que acabamos de leer del disco es un RETURN (lo que indica que hemos terminado de leer una línea del programa) H$ se imprime y el proceso continúa hasta que ha terminado el programa. (La línea 150 es otro resto.)

La subrutina 170 creemos que se explica por sí sola. No obstante, por razones de presentación, hemos tenido que usar la terminología inglesa para designar los movimientos del cursor al no poder distinguir con una sola letra y sin el uso de gráficos las funciones ARRIBA-ABAJO.

(continúa en la pág. siguiente)

los listados de CLUB COMMODORE

(conclusión)

Otro «resto» de pruebas es la línea 350, que sin el REM, iniciaba una nueva línea de listado cada vez que se encontraba un signo ":" Así separaba las sentencias de una línea, pero los resultados no justificaban, a nuestro juicio, el «alargamiento» del listado.

NUEVO TIPO DE LISTADO

```
PROGRAMA LIST DE LIST

READY.

 0 REM PROGRAMA PARA HACER LISTADOS (CBM 80
23)
 10 H$="":X=0
 20 PRINT"[ CLR ]PROGRAMA PARA LISTAR PROGR
AMAS"
 30 INPUT"NOMBRE";S$
 40 INPUT"CARACTERES";B
 50 OPEN1,9,2,"0:"+S$+",S,R":OPEN4,4:PRINT#
4,CHR$(0)
 51 OPEN13,4,13:PRINT#13:CLOSE13
 52 OPEN15,4,15:PRINT#15:CLOSE15
 60 PRINT#4,"PROGRAMA "+S$
 70 PRINT#4,CHR$(1)
 100 IFST=64THENCLOSE1:PRINT#4:CLOSE4:RUN
 110 GET#1,A$
 120 GOSUB170
 130 H$=H$+A$
 140 IF(LEN(H$)>=B)OR(A$=CHR$(13))THENPRINT
#4,H$:PRINTH$:H$=""
 150 REMIFLEN(H$)=BTHENPRINT#4,H$:PRINTH$:H
$=""
 160 GOTO100
 170 IFA$="[CRSRD]"THENA$="[CRSRD]":RETURN
 180 IFA$="[CRSRU]"THENA$="[CRSRU]":RETURN
 190 IFA$="[CRSRR]"THENA$="[CRSRR]":RETURN
 200 IFA$="[CRSRL]"THENA$="[CRSRL]":RETURN
 210 IFA$="[HOME ]"THENA$="[HOME ]":RETURN
 220 IFA$="[ CLR ]"THENA$="[ CLR ]":RETURN
 230 IFA$="[ BLK ]"THENA$="[ BLK ]":RETURN
 240 IFA$="[ WHT ]"THENA$="[ WHT ]":RETURN
 250 IFA$="[ RED ]"THENA$="[ RED ]":RETURN
 260 IFA$="[ CYN ]"THENA$="[ CYN ]":RETURN
 270 IFA$="[ PUR ]"THENA$="[ PUR ]":RETURN
 280 IFA$="[ GRN ]"THENA$="[ GRN ]":RETURN
 290 IFA$="[ BLU ]"THENA$="[ BLU ]":RETURN
 300 IFA$="[ YEL ]"THENA$="[ YEL ]":RETURN
 310 IFA$="[RVSON]"THENA$="[RVSON]":RETURN
 320 IFA$="[RVSOF]"THENA$="[RVSOF]":RETURN
 330 IFA$="[INST ]"THENA$="[INST ]":RETURN
 340 IFA$="[CTRL←]"THENA$="[CTRL←]":RETURN
 350 REMIFA$=":"THENA$=":"+CHR$(13)
 360 IFA$="[COMM+]"THENA$="[COMM+]":RETURN
 370 IFA$="[SHIF+]"THENA$="[SHIF+]":RETURN
 380 IFA$="[COMM-]"THENA$="[COMM-]":RETURN
 390 IFA$="[SHIF-]"THENA$="[SHIF-]":RETURN
 400 IFA$="[COMM£]"THENA$="[COMM£]":RETURN
 410 IFA$="[SHIF£]"THENA$="[SHIF£]":RETURN
 420 IFA$="[COMMQ]"THENA$="[COMMQ]":RETURN
 430 IFA$="[SHIFQ]"THENA$="[SHIFQ]":RETURN
 440 IFA$="[COMMW]"THENA$="[COMMW]":RETURN
 450 IFA$="[SHIFW]"THENA$="[SHIFW]":RETURN
 460 IFA$="[COMME]"THENA$="[COMME]":RETURN
 470 IFA$="[SHIFE]"THENA$="[SHIFE]":RETURN
 480 IFA$="[COMMR]"THENA$="[COMMR]":RETURN
 490 IFA$="[SHIFR]"THENA$="[SHIFR]":RETURN
 500 IFA$="[COMMT]"THENA$="[COMMT]":RETURN
 510 IFA$="[SHIFT]"THENA$="[SHIFT]":RETURN
 520 IFA$="[COMMY]"THENA$="[COMMY]":RETURN
 530 IFA$="[SHIFY]"THENA$="[SHIFY]":RETURN
 540 IFA$="[COMMU]"THENA$="[COMMU]":RETURN
 550 IFA$="[SHIFU]"THENA$="[SHIFU]":RETURN
 560 IFA$="[COMMI]"THENA$="[COMMI]":RETURN
 570 IFA$="[SHIFI]"THENA$="[SHIFI]":RETURN
 580 IFA$="[COMMO]"THENA$="[COMMO]":RETURN
 590 IFA$="[SHIFO]"THENA$="[SHIFO]":RETURN
 600 IFA$="[COMMP]"THENA$="[COMMP]":RETURN
 610 IFA$="[SHIFP]"THENA$="[SHIFP]":RETURN
 620 IFA$="[COMM@]"THENA$="[COMM@]":RETURN
 630 IFA$="[SHIF@]"THENA$="[SHIF@]":RETURN
 640 IFA$="[COMM*]"THENA$="[COMM*]":RETURN
 650 IFA$="[SHIF*]"THENA$="[SHIF*]":RETURN
 660 IFA$="[COMM↑]"THENA$="[COMM↑]":RETURN
 670 IFA$="[COMM↑]"THENA$="[SHIF↑]":RETURN
 680 IFA$="[COMMA]"THENA$="[COMMA]":RETURN
 690 IFA$="[SHIFA]"THENA$="[SHIFA]":RETURN
 700 IFA$="[COMMS]"THENA$="[COMMS]":RETURN
 710 IFA$="[SHIFS]"THENA$="[SHIFS]":RETURN
 720 IFA$="[COMMD]"THENA$="[COMMD]":RETURN
 730 IFA$="[SHIFD]"THENA$="[SHIFD]":RETURN
 740 IFA$="[COMMF]"THENA$="[COMMF]":RETURN
 750 IFA$="[SHIFF]"THENA$="[SHIFF]":RETURN
 760 IFA$="[COMMG]"THENA$="[COMMG]":RETURN
 770 IFA$="[SHIFG]"THENA$="[SHIFG]":RETURN
 780 IFA$="[COMMH]"THENA$="[COMMH]":RETURN
 790 IFA$="[SHIFH]"THENA$="[SHIFH]":RETURN
 800 IFA$="[COMMJ]"THENA$="[COMMJ]":RETURN
 810 IFA$="[SHIFJ]"THENA$="[SHIFJ]":RETURN
 820 IFA$="[COMMK]"THENA$="[COMMK]":RETURN
 830 IFA$="[SHIFK]"THENA$="[SHIFK]":RETURN
 840 IFA$="[COMML]"THENA$="[COMML]":RETURN
 850 IFA$="[SHIFL]"THENA$="[SHIFL]":RETURN
 860 IFA$="[COMMZ]"THENA$="[COMMZ]":RETURN
 870 IFA$="[SHIFZ]"THENA$="[SHIFZ]":RETURN
 880 IFA$="[COMMX]"THENA$="[COMMX]":RETURN
 890 IFA$="[SHIFX]"THENA$="[SHIFX]":RETURN
 900 IFA$="[COMMC]"THENA$="[COMMC]":RETURN
 910 IFA$="[SHIFC]"THENA$="[SHIFC]":RETURN
 920 IFA$="[COMMV]"THENA$="[COMMV]":RETURN
 930 IFA$="[SHIFV]"THENA$="[SHIFV]":RETURN
 940 IFA$="[COMMB]"THENA$="[COMMB]":RETURN
 950 IFA$="[SHIFB]"THENA$="[SHIFB]":RETURN
 960 IFA$="[COMMN]"THENA$="[COMMN]":RETURN
 970 IFA$="[SHIFN]"THENA$="[SHIFN]":RETURN
 980 IFA$="[COMMN]"THENA$="[COMMN]":RETURN
 990 IFA$="[COMMN]"THENA$="[COMMM]":RETURN
 1000 IFA$="[SHIFM]"THENA$="[SHIFM]":RETURN
 1010 RETURN
READY.
```

MULTICENTRO DE INFORMATICA

Chips & Tips

PUERTO RICO, 21-23 - MADRID-16 TEL. 250 74 02 - 250 74 04

commodore VIC-20

JUEGOS

- COMECOCOS. 3,5K. A/R. G/E. JY. EXTRAORDINARIA VERSION DEL POPULAR PUCKMAN. COLOR Y MOVILIDAD INSUPERABLE . 1.900
- VICGAMON. +3K. JUEGO DE INTELIGENCIA QUE LE MANTENDRA EN TENSION HASTA DERROTAR A SU VIC 1.800
- ASTEROIDS WAR. 3,5K. C/M. A/R. G/E. JY. ESPECTACULAR BATALLA GALACTICA CONTRA LA NUBE PROTONICA EN 3 DIMENSIONES 1.800
- FROGGER. +3K y 3,5K. C/M. A/R. G/E. JY. ULTIMA NOVEDAD EN EE.UU. CRUZAR EL PELIGROSO RIO Y LA AUTOPISTA SUICIDA 2.000
- RATMAN. +8K. C/M. A/R. G/E. DE LA BOVEDA CELESTE DESCENDERAN EXTRAÑAS RATAS ATOMICAS. ESPECTACULAR ANIMACION 1.900
- SHARK ATTACK. 3,5K. C/M. A/R. JY. EN MEDIO DEL OCEANO SERA ATACADO POR LOS PELIGROSOS TIBURONES. DEFIENDASE CON SU RED ATOMICA.................................... 1.900
- ROX III. 3,5K Y+8K. C/M.A/R. G/E. JY. DESDE SU SOFISTICADA BASE LUNAR DEFIENDA SU PLANETA DEL ATAQUE DE LOS UFOS 1.800
- ,ULTISOUND SYNTHETIZER. 3,5K. ¿UN ORGANO EN SU VIC? ¿CON ACOMPAÑAMIENTO, BATERIA Y EFECTOS ESPECIALES? . . 1.900
- SKI-RUN. 3,5K. C/M. A/R. G/E. DESLICESE POR LAS HELADAS PISTAS DE COMPETICION. SLALOM, S/GIGANTE, DESCENSO. 9 NIVELES 1.800
- FIREBIRD. (SPACE PHREEKS). 3,5K. C/M. A/R. G/E. JY. AÑO 3.010. VD. ES EL UNICO SUPERVIVIENTE DE LA BATALLA DE RIGELLIAN. DEBERA COLONIZAR OTRO PLANETA Y LUCHAR CONTRA LAS CRIATURAS GALACTICAS..................... 1.900
- BREAKOUT. 3,5K. CONSIGA DESTRUIR LA PARED DE LADRILLOS MULTICOLORES CON LA BOLA MAGICA. INCLUYE "MASTERMIND". 1.600
- AJEDREZ. PRIMERA VERSION EN CASSETTE CON GRAFICOS EN ALTA RESOLUCION. BASTANTES NIVELES DE JUEGO. (STANDARD) . 2.800
- SHADOWFAX. INCREIBLES GRAFICOS ANIMADOS. EL CABALLERO DE LAS SOMBRAS EN LUCHA CONTRA LOS JINETES DEL TIRANO INVASOR. (STANDARD)................................... 1.900
- SNAKE. COLORIDO, MOVIMIENTOS Y GRAFICOS EXCEPCIONALES. VERSION DEL FAMOSO JUEGO DE LAS SERPIENTES (SNAKE). (STANDARD)... 1.900
- MYRIAD. +3K. C/M. A/R. G/E. JY. LA MAS ESPECTACULAR AERONAVE PARA DESTRUIR LAS CRIATURAS COSMICAS EN SU VIAJE GALACTICO... 2.000
- COSMIADS. 3,5K. C/M. A/R. G/E. JY. VERSION ULTRARRAPIDA DEL MUNDIALMENTE FAMOSO JUEGO "GALAXIANS". INCREIBLES EFECTOS SONOROS....................................... 1.700
- BLITZRIEG. 3,5K. C/M. A/R. G/E. JY. DESTRUYA LA CIUDAD ENEMIGA CON SU BOMBARDERO. 25 NIVELES DE JUEGO 1.600
- DEFENSA. +8K. C/M. A/R. G/E. JY. N.º 1 EN INGLATERRA. COMO GUERRERO GALACTICO DEBE DEFENDER A LOS HUMANOIDES CONTRA LOS ENEMIGOS CIBERNETICOS. 9 NIVELES DE JUEGO. ESPECTACULAR NAVE Y SONIDOS 2.000
- VIC PANIC. 3,5K. C/M. A/R. G/E. JY. VERSION DEL POPULAR "SPACE PANIC". ESCALE LAS LADERAS Y HUYA DE LOS MONSTRUOS 1.800
- SKRAMBLE. 3,5K. C/M. A/R. G/E. JY. ATRAVESANDO LOS TEMIBLES PASADIZOS INTERESTELARES DESTRUYA LAS BASES ENEMIGAS 1.900
- 3D LABYRINTH. +8K. C/M. A/R. EXTRAORDINARIO LABERINTO TRIDIMENSIONAL. ¿SERA CAPAZ DE SALIR DE EL? UNO O VARIOS JUGADORES 1.800
- GOLF. 3,5K. RECORRIDO DE 9 HOYOS PERO ATENCION A LOS OBSTACULOS: ARBOLES, LAGOS, ETC. INCLUYE VIC MUSIC Y PIANO 1.600
- CARRERA DE BUGGYS. 3,5K. C/M. A/R. G/E. ESPECTACULAR RECORRIDO. ACELERADOR. DECELERACION. 9 NIVELES 1.800
- GRIDRUNNER. 3,5K. C/M. A/R. G/E. JY. IMPRESIONANTE VERSION LLENA DE COLORIDO, MOVILIDAD Y SONIDO DEL POPULAR "CENTIPEDE"... 1.900
- HI-RES. 3,5K. GRAN JUEGO DEMO/UTILIDAD PARA REALIZAR EN PANTALLA GRAFICOS EN ALTA RESOLUCION. INCLUYE GEN. CARACTERES... 1.500
- ABDUCTOR. LAS CRIATURAS COSMICAS DEL PLANETA "ALPHA I" INTENTARAN SECUESTRAR A LOS HUMANOIDES PARA CONSEGUIR ENERGIA E INTELIGENCIA SUPERIORES. TU MISION SERA DEFENDER TU PLANETA Y DESTRUIR LAS NAVES ABDUCTORAS. (STANDARD). 1.800
- TRAXX. VERSION DEL CONOCIDO JUEGO "AMIDAR"; MEZCLA DEL POPULAR "PACKMAN" Y DEL JUEGO "QUIX". 100% CODIGO MAQUINA. GRAFICOS EN ALTA RESOLUCION. ESPECTACULAR SONIDO Y COLOR. 8K DE MEMORIA 2.000

UTILIDADES

- VIC PRINT. +8K. EXTRAORDINARIO Y SENCILLO PROCESADOR DE TEXTOS. TABULACION, MAQUETACION, CABECERAS, COPIAS. CASS O DISK... 2.000
- VIC LABEL. +8K. EN COMBINACION CON VIC PRINT, ELABORA ETIQUETAS PARA DIRECCIONES............................ 1.900
- VIC POST. +8K. ELABORA LETRAS Y TEXTOS ESPECIALES EN TAMAÑO Y FORMA PARA POSTERS, LISTAS DE PRECIOS, ETC. . . 2.900
- VIC CALC. HERRAMIENTA DE CALCULO QUE SUSTITUYE AL LAPIZ, PAPEL Y CALCULADORA, REALIZA COMPLEJOS MODELOS FINANCIEROS CON POSIBILIDAD DE AJUSTARLO A OTROS PARAMETROS CON SOLO PULSAR UNA TECLA. 16K DE MEMORIA............ 3.200
- VIC BASE. 16K. POTENTE BANCO DE DATOS. 255 CARACTERES, MAS DE 25 CAMPOS. CAMBIO Y LOCALIZACION, SALIDA IMPRESORA.. 3.200
  OPCION 3,5K... 1.800
- GRAPHVICS. +3K. AÑADE 18 POTENTES COMANDOS PARA POSICIONAR PUNTOS, DIBUJAR LINEAS Y TEXTOS EN ALTA RESOLUCION (152×160) 2.200
- GRAPH EDITOR & SOFTKEY 24, 3,5K. AMBOS PROGRAMAS PERMITEN DISEÑAR HASTA 64 CARACTERES PARA INCORPORARLOS A SUS PROPIOS PROGRAMAS Y JUEGOS 2.000

EDUCATIVOS

- QUIZ-MASTER. +3K. EL MAS ESPECTACULAR AVANCE EDUCATIVO. PERMITE LA CORRECCION Y PUNTUACION DE TODAS LAS RESPUESTAS QUE RECIBE EL ORDENADOR......................
- QUIZ SET-UP. EN TANDEM CON QUIZ—MASTER PERMITE LA ELABORACION POR EL USUARIO DE TODO TIPO DE PREGUNTAS Y CUESTIONES EDUCATIVAS O DE ENTRETENIMIENTO, EGB, IDIOMAS, MATEMATICAS, HISTORIA, GEOGRAFIA, ETC. CREANDO UN AGIL Y ATRACTIVO SISTEMA DOMESTICO/EDUCATIVO .....

  (QUIZ-MASTER + QUIZ SET-UP) 3.200
- FACEMAKER. 16K. CARICATURANDO EL ROSTRO DE SUS COMPAÑEROS Y AMIGOS EL VIC 20 PONDRA A PRUEBA EL VOCABULARIO Y LA ATENCION DEL NIÑO.................................. 2.000
- NUMBER CHASER. 16K. PROGRAMA PARA PRACTICAS DE MULTIPLICACION CON CARRERAS DE COCHES, ADELANTA, FRENA, ACELERA SEGUN LAS RESPUESTAS. 4 NIVELES DE DIFICULTAD . . . 2.000
- NUMBER GULPER. 16K. JUEGO EDUCACIONAL DE COMPETICION CON NUMEROS PARA SUMA, RESTA, MULTIPLICACION Y DIVISION 2.000
- WE WANT TO COUNT. 16K PROGRAMA PARA NIÑOS A PARTIR DE TRES AÑOS, INVASORES, CARRERAS, ETC. 2.000
- TWISTER. 16K. JUEGO DE LOGICA Y CONCENTRACION. PUZZLES GEOMETRICOS CON SONIDO Y COLOR............................ 2.000

LIBROS

VIC REVEALED	2.200	ASSEMBLER	2.000	ZAP! POW! BOOM!	1.800
GETTIN ACQUAINTED WITH YOUR VIC 20	1.800	SYNPHONY MELANCHOLY COMP.	1.800	VIC INNOVATIVE	2.000
50 PROGRAMAS LISTADOS I	1.500	50 PROGRAMAS LISTADOS II	1.500	50 PROGRAMAS LISTADOS III	1.500

ALMACENAMIENTO DE DATOS



grabación de datos en cassette

por **P. MASATS**

Muchos usuarios nos dirigen consultas respecto a la utilización de cassettes en el VIC. Se quejan de que no pueden escribir sus datos en el cassette y volverlos a leer, por lo cual dedicamos este artículo a los sistemas de grabación y lectura de datos en cinta. Es posible hacer cosas más sofisticadas de las que vamos a describir pero lo que aquí se describe puede servir de punto de partida para la experimentación.

Escribir en el cassette es como escribir en la pantalla excepto que la información queda inscrita en la cinta como series de impulsos.

Antes de escribir nada en el cassette, debe abrirse un área de trabajo utilizando OPEN2,1,1,"NOMBRE DEL FICHERO". Esto le indica al VIC que queremos utilizar el cassette y empezará la operación grabando en la cinta el título (en este caso, "NOMBRE DEL FICHERO"). Esta operación es vital puesto que marca el principio del fichero (el fichero es el conjunto de datos que vamos a grabar bajo la denominación "NOMBRE DEL FICHERO"), y también asegura que se ha puesto en marcha el aparato de cassette. Téngase en cuenta que el VIC puede advertir cuándo se ha pulsado una tecla en el cassette pero no distingue entre PLAY, RECORD, F.FWD o REW. De manera que depende del usuario el asegurarse de que ha apretado las teclas correctas al recibir el mensaje "PRESS RECORD & PLAY ON TAPE".

El primer número después de OPEN dice cómo nos vamos a referir a los datos durante el resto del programa y puede ser un número arbitrario entre 1 y 127. Este número designa el fichero lógico con el que se va a trabajar. Entendemos por fichero lógico una zona de memoria (que se asigna por el sistema operativo y que suele ser de 192 bytes) que sirve de intermedio entre el ordenador y el periférico —el cassette en nuestro caso—. Su función es servir de almacenamiento temporal de los datos hasta completar un bloque de dimensiones determinadas por la naturaleza del periférico que se vaya a utilizar, dado que en la cinta se graba un paquete de 192 bytes. En este «BUFFER» (es el nombre que se suele dar a estas zonas de memoria) se van guardando los datos generados por el programa hasta tener la cantidad necesaria. Entonces se graba y se vuelve a repetir la operación hasta que el programa encuentra la sentencia CLOSE y, entendiendo que se han terminado los datos a grabar, envía los que tiene aunque no esté lleno. En este caso utilizamos el fichero lógico número 2, y utilizaremos PRINT # 2 para acceder a la unidad de cassette. La razón por la que esto es necesario es que, con discos e impresoras, podemos acceder a varios canales de datos diferentes en el curso de un mismo programa.

El segundo número identifica al periférico que queremos utilizar (el nú-

mero 1 es la unidad de cassette) y el tercero se utiliza para seleccionar lectura(0) o escritura(1). La interpretación de este tercer número depende del periférico utilizado.

Finalmente, los caracteres entre comillas designan el título del fichero de datos que vamos a grabar. Esto es útil a la hora de leer para asegurarnos de que vamos a leer el fichero correcto.

Durante el comando OPEN, la cinta se moverá un momento y luego se parará. El cassette está entonces listo para recibir datos. En vez de PRINT, utilizamos PRINT# 2, lo cual envía el dato al periférico especificado en el comando de OPEN. Nótese la coma entre PRINT# 2 y la variable.

En las primeras etapas de práctica en el uso del cassette es mejor tener sólo una variable para cada PRINT# 2 utilizado y asegurarse de que la instrucción no finaliza en coma o punto y coma. Al leer el dato, necesitamos un «return» al final de cada variable al igual que hemos de apretar RETURN cuando tecleamos datos en el teclado. Esto se escribirá en el cassette al final del dato, suponiendo que no le hemos indicado lo contrario con una coma o un punto y coma.

Una vez se han escrito todos los datos, es esencial decirle al VIC que hemos acabado nuestro trabajo con la unidad de cassette. Esto se hace mediante el comando CLOSE, que se asegurará de que los últimos datos están en la cinta (sería posible que aún estuvieran en el VIC esperando a ser escritos). En nuestro ejemplo utilizamos CLOSE2 para acabar, de manera que el número utilizado sea el mismo que el primer número en el comando OPEN. Para operaciones seguras con cassette no está nunca de más insistir en la importancia del CLOSE.

Imaginemos que hemos preparado dos «arrays» en memoria. Cada uno de ellos, con 20 valores —NOM$(I), contiene los nombres de los amigos, y NUM(I) contiene sus números de teléfono. Si queremos grabar esta información en cassette, podemos hacerlo como sigue:

```
1000 OPEN2,1,1,"LISTIN"
1010 FORI=1TO20
1020 PRINT# 2,NOM$(I)
1030 PRINT# 2,NUM$(I)
1040 NEXTI
1050 CLOSE2
```

Ahora tenemos que volver a leer los datos. Después de rebobinar el cassette, utilizamos de nuevo el OPEN pero ahora en la forma OPEN2, 1,0,"LISTIN", donde el cero indica que queremos leer los datos. También incluimos el nombre con que grabamos el fichero; y el VIC empezará a leer el cassette, buscando el título con que hayamos hecho el OPEN.

Si el nombre se omite y utilizamos OPEN2,1,0 (o incluso OPEN2, puesto que el VIC asume que queremos una lectura en el cassette si estos parámetros no se dan) el VIC trabajará con el primer título del cassette.

Una vez encontrado el fichero, los datos están listos para leerse. Utilizamos simplemente INPUT# 2 en lugar de INPUT, y nos aseguramos de que leemos los datos en el mismo orden en que han sido escritos, aunque los nombres de las variables no tienen por qué ser los mismos. O sea que para leer la lista de teléfonos, podemos utilizar:

```
1000 OPEN2,1,0,"LISTIN"
1010 FORI=1TO20
1020 INPUT# 2,A$(I)
1030 INPUT# 2,B$(I)
1040 NEXTI
1050 CLOSE2
```

Utilizar un PRINT# con diferentes variables, de la misma manera en que lo haríamos en la pantalla, puede causar problemas, pues el cassette no tiene manera de identificar dónde acaba una variable (la parte correspondiente a cada variable) y dónde empieza la siguiente. Debe haber comas entre estas variables y el INPUT # debe tener el mismo número de variables que el PRINT# correspondiente. Hasta que se esté totalmente familiarizado con estos problemas, es aconsejable evitar tener más de una variable en cada instrucción PRINT#.

Igualmente, debe tenerse cuidado en evitar leer más allá del último dato escrito — si es necesario utilícese un valor especial para señalizar el final de los datos (p. ej., 9999) que le permita detener el programa en cuanto se lea. Leer más allá del último dato escrito puede causar problemas y obtener un mensaje "STRING TOO LONG ERROR", incluso si lo que se estaba buscando era un número.

(pasa a la pág. siguiente)

grabación de datos en cassette

(conclusión)

Si se tienen dudas sobre el contenido de un fichero en concreto, es posible extraer los datos grabados carácter a carácter y verlos a través de la pantalla con el programa siguiente:

```
100 OPEN2
110 GET# 2,A$
120 PRINTA$
130 GOTO110
```

Depurar programas utilizando el cassette es, a veces, bastante difícil, y hay un truco que permite poner los datos en la pantalla, en vez de en el cassette, para asegurarse de que la estructura es la correcta, sin convertir el PRINT# 2 en PRINT. Si se cambia el OPEN del periférico 1 al periférico 3 (la pantalla) — en otras palabras, OPEN2,3,1,"NOMBRE DEL FICHERO" — todos los datos aparecerán en la pantalla, en lugar de en el cassette.

Igualmente es posible imitar la lectura del cassette utilizando el periférico 0 (el teclado) en el comando OPEN, y simular el INPUT# del cassette tecleando Ud. mismo los datos en el teclado. En este caso le advertimos de que no recibirá el «?» normal. Aparecerá simplemente un cursor destelleante.

Debido a la manera en que el cassette graba los datos no es posible escribir paquetes individuales de datos para actualizarlos. En cambio, todos los datos deben leerse dentro del VIC, realizar los cambios necesarios y luego grabarse nuevamente en su totalidad. Esto puede resultar algo tedioso pero es la única manera. La totalidad de los datos registrados en la cinta desde el OPEN hasta el CLOSE se conoce como el fichero de datos.

Puede haber diversos ficheros en el cassette, cada uno con un NOMBRE o título diferente. Si el VIC no encuentra el fichero especificado en el comando OPEN de lectura, continuará leyendo la cinta hasta encontrar el que está buscando. Esto puede suponer un problema si sólo se tienen uno o dos ficheros en el programa y el VIC procede a leer la totalidad de una cinta C90 para un fichero que no existe.

Para evitar leer más allá del último fichero, es posible etiquetarlo como el último en la cinta. Para ello utilizamos OPEN2,1,2,"NOMBRE DEL FICHERO".

En otras palabras, sustituimos el 1 (que indica que queremos escribir un dato) por el 2 (que indica que queremos escribir pero que es el último dato que aparecerá en el cassette). Si el cassette se lee y el fichero que buscamos no se encuentra, obtendremos un error de "FILE NOT FOUND", y la búsqueda finalizará.

Dado que el cassette que utilizamos es idéntico a los cassettes normales de audio, es posible escuchar la cinta en un reproductor de música. En primer lugar se escucharán unos nueve segundos de una señal de audio pura. Es la señal que sirve para sincronizar la operación de lectura. A continuación, tres segundos de zumbido, que son el título de bloque. El dato se graba en bloques de 192 caracteres (que es la razón por la cual el cassette se para y se pone en marcha mientras está en uso) y cada uno es precedido por una señal de sincronización y seguido de un zumbido que es el dato. Dado que cada bloque está escrito dos veces por razones de seguridad, habrá una pausa allí donde empieza la segunda copia.

Un programa tiene la misma estructura pero está grabado como un largo bloque cuya longitud depende de la longitud del propio programa aunque seguirá habiendo una pausa allí donde empiece la segunda copia del mismo.

Si el cassette está totalmente rebobinado, hay una gran probabilidad de que haya hasta diez segundos de cinta transparente que no graba, al principio de la cinta. Si este trozo es de menos de siete u ocho segundos, no debería haber ningún problema. Pero si es demasiado larga, la señal de sincronismo, o incluso el bloque con el título, podrían no quedar escritos en la cinta del cassette. Ésta es probablemente la principal causa de los fallos del cassette en lectura/escritura puesto que la lectura no puede ser realizada sin esa señal de sincronización y el bloque del título. La solución es asegurarse de que el cassette está rebobinado sólo hasta el final de la cinta transparente cada vez que tengamos que grabar datos. ■

BANCO DE PRUEBAS

el cartucho de ajedrez del VIC-20 (y II)

por **PERE MASATS**

Cuando el VIC está «pensando» una jugada, el logotipo de COMMODORE aparece en color rojo. También puede verse el número de movimiento futuro que el VIC está estudiando en la parte derecha de la pantalla. Asimismo el movimiento que la máquina está considerando se puede ver en la parte inferior del registro de movimientos.

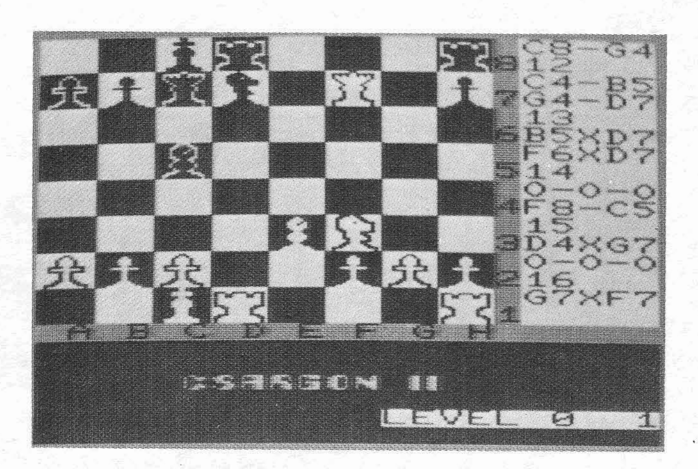
Una característica interesante para neófitos en ajedrez es la de que, jugando en niveles superiores al 0, al llegar el turno del operador si éste pulsa las teclas SHIFT y/(el signo «?») el VIC aconsejará al usuario el próximo movimiento escribiéndolo en la parte inferior del registro de movimientos. Si el jugador decide aceptar el consejo, simplemente debe pulsar RETURN y la indicada por el VIC será su próxima jugada. (Si uno se fía demasiado de estos consejos su ego puede salir perjudicado, pues el VIC tiende a «barrer para casa».)

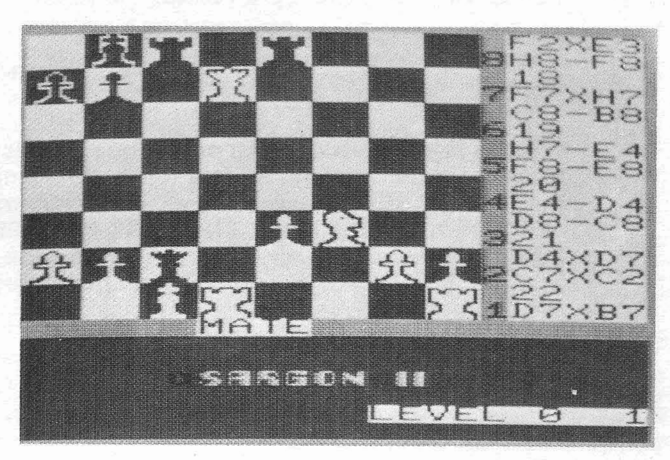
Para jugar normalmente se deben entrar las coordenadas de la pieza que se desea mover y las de la casilla de destino separadas por un signo «-». También es posible jugar con la palanca para juegos (JOYSTICK). Para ello se debe situar sobre la pieza que se desea mover y se debe pulsar el botón rojo. Seguidamente se mueve el cursor sobre el lugar de destino y se pulsa el botón rojo. También pueden realizarse movimientos especiales: ENROQUE: solamente es necesario entrar el movimiento del rey, el resto se realiza automáticamente; captura EN PASSANT se debe entrar el movimiento del peón.

Si se ha escogido el modo SETUP, aparecerá el tablero de ajedrez con la casilla inferior izquierda parpadeando, con las teclas de movimiento del cursor se sitúa sobre la casilla que se quiere modificar y se procede como sigue:

—Si se desea eliminar una pieza que está en esta casilla, se pulsa la tecla INST/DEL.

—Si se necesita poner una pieza en dicha casilla, se debe teclear un código que es como sigue: K = rey, Q = reina, R = torre, B = álfil, N = caballo y P = peón. A continuación de esta letra debe indicarse el color de la pieza: B = negra y W = blanca. Por último, un número indicará al VIC si esta pieza se supone que ha sido previamente movida o no: 0 si no ha sido movida y 1 si lo ha sido con anterioridad.





Cuando se ha completado la configuración del tablero se debe pulsar RETURN y el VIC preguntará: COLOUR TO MOVE NEXT (B,W): es decir ¿qué color es el próximo en mover? A lo que se contestará con el código ya conocido. Seguidamente el equipo pregunta: MOVE NUMBER (NN): es decir, el número de movimiento en el que se establece el juego. Este número se considera por el programa para establecer la estrategia según el desarrollo de la partida. A continuación: YOUR COLOR (B,W): y LEVEL OF PLAY (0-6): que son los mismos que en el juego normal.

Como puede comprobarse fácilmente, las características de este programa lo hacen muy interesante, tanto para el aficionado experto en ajedrez como para el principiante. Como ilustración en las figuras se incluye una breve partida.

A título meramente indicativo, los tiempos que invierte el VIC en sus movimientos en función del nivel de juego son los siguientes:

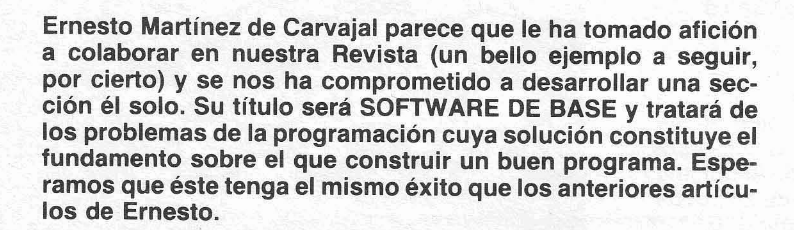
0: Respuesta inmediata
1: 20 segundos
2: 1 minuto
3: 2 minutos
4: 6 minutos
5: 40 minutos
6: 4 horas

Estos tiempos son muy variables y pueden resultar variaciones entre un tercio y tres veces de los valores dados anteriormente. ■

SOFTWARE DE BASE (I)

técnicas de ordenación en memoria

por E. MARTÍNEZ DE CARVAJAL



Ernesto Martínez de Carvajal parece que le ha tomado afición a colaborar en nuestra Revista (un bello ejemplo a seguir, por cierto) y se nos ha comprometido a desarrollar una sección él solo. Su título será SOFTWARE DE BASE y tratará de los problemas de la programación cuya solución constituye el fundamento sobre el que construir un buen programa. Esperamos que éste tenga el mismo éxito que los anteriores artículos de Ernesto.

Uno de los más típicos problemas con que se enfrentan los ordenadores es la clasificación de datos y, por lo tanto, los «sorts» son uno de los principales elementos constituyentes de la familia denominada software de base, tanto en las aplicaciones comerciales o de gestión como científicas, por lo que se han desarrollado muchos tipos que, si bien tienen un fin común, cubren las diferentes necesidades de cada usuario. Los factores a tener en cuenta a la hora de escoger un método de ordenación son: velocidad, recursos necesarios, longitud de la rutina y sencillez de la programación. Los dos primeros son los de más peso. Teniendo en cuenta que el VIC es uno de los micros más rápidos del mercado, se le puede sacar bastante partido en este campo.

Con este artículo inicio una serie destinada a describir y comparar los diferentes medios de ordenación más usuales. En términos generales se pueden agrupar en dos grandes superfamilias que se acostumbran a denominar «métodos comparativos» y «métodos distributivos». En la primera de ellas, se considera la clave (es la parte del elemento que se utiliza) como un todo en la comparación mientras que en la segunda se considera la clase carácter a carácter o como entidades separadas y, entonces, la colocación relativa de un elemento durante la ordenación no depende de la magnitud de su clave sino de sus características con relación a un modelo dado. Un ejemplo de la primera puede ser la ordenación alfabética de una serie de nombres y, de la segunda, la ordenación de una baraja de cartas que primero habría que separar por palos y después, dentro de éstos, por cifras.

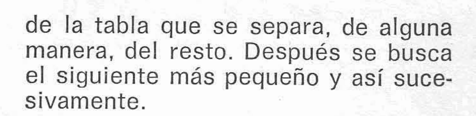
Dentro de la primera superfamilia se pueden considerar las siguientes familias:

MÉTODO DE ENUMERACIÓN O CONTAJE

La idea de este método es muy sencilla y se presta muy bien a aplicarla manualmente a un grupo de cifras en un papel. Consiste en contar el número de elementos inferior a cada uno de ellos. Al final, el valor que nos indique cada uno de los contadores más uno, nos dará la posición relativa del elemento dentro de la tabla. Sólo se requiere un espacio adicional para un contador por cada elemento de la tabla y un tiempo de ordenador que no es tan pequeño a medida que crece la tabla.

MÉTODO DE SELECCIÓN

La base de este método consiste en localizar el elemento más pequeño



de la tabla que se separa, de alguna manera, del resto. Después se busca el siguiente más pequeño y así sucesivamente.

MÉTODO DE INTERCAMBIO

La idea de este método es ir recorriendo la tabla a ordenar e ir intercambiando entre sí cada pareja de elementos desordenados, repitiendo el proceso hasta que no haya ningún par de éstos. Es uno de los métodos más utilizados, cuando no es muy importante el tiempo de ordenación, ya que une la sencillez de programación a los pocos recursos adicionales de memoria.

MÉTODO DE INSERCIÓN

Este método se basa en la inserción sucesiva de nuevos elementos en una tabla que va creciendo ordenada. Es decir, se toma un elemento cada vez y se inserta en su posición correcta respecto de los elementos ya clasificados.

MÉTODO DE FUSIÓN

La idea de este método es tratar la tabla inicial como trozos que se tienen que intercalar, existiendo tantos trozos como rupturas de secuencia haya o bien como **N** trozos de **I** elementos cada uno. A base de fusiones sucesivas, se obtienen trozos cada vez más grandes hasta llegar a tener la lista totalmente ordenada.

Paso ahora a detallaros cada uno de estos métodos. Este artículo lo dedicaremos al primero de ellos:

CLASIFICACIÓN POR MONTAJE O ENUMERACIÓN

Éste es un método con poco porvenir, mejor sería decir ninguno, pudiéndose considerar como una vía muerta

```
10 REM SUBRUTINA SORT POR NUMERACION O CONTA
JE
20 REM --------------------------------------
--
30 REM 01:82 - ERNESTO MTNZ. DE CARVAJAL HED
RICH
40 INPUT "NUM. ELEMENTOS : ";N
50 DIM T1(N),T2(N),T3(N)
60 FOR I=1TON:T1(I)=INT(RND(0)*100):NEXTI
80 TI$="000000"
100 FOR I=1 TO N-1
110 FOR J=I+1 TO N
120 IF T1(I)>T1(J) THEN T2(I)=T2(I)+1:GOTO 1
40
130 T2(J)=T2(J)+1
140 NEXTJ
150 NEXTI
155 FORI=1TON:T3(T2(I)+1)=T1(I):NEXTI
160 T$=TI$
170 PRINT "TIEMPO ORDENACION PARA ";N;" ELEM
ENTOS : ";MID$(T$,3,2);":";MID$(T$,5,2)
READY.
```

```
10 REM SUBRUTINA SORT POR NUMERACION O CONTA
JE
20 REM --------------------------------------
--
30 REM 01:82 - ERNESTO MTNZ. DE CARVAJAL HED
RICH
40 INPUT "NUM. ELEMENTOS : ";N
50 DIM T1(N),T2(N),T3(N)
60 FOR I=1TON:T1(I)=INT(RND(0)*100):NEXTI
80 TI$="000000"
90 REM ==============
100 FOR I=1 TO N-1
110 FOR J=I+1 TO N
120 IF T1(I)>T1(J) THEN T2(I)=T2(I)+1:GOTO 1
40
130 T2(J)=T2(J)+1
140 NEXTJ
150 NEXTI
155 FORI=1TON:T3(T2(I)+1)=T1(I):NEXTI
157 REM ==============
160 T$=TI$
170 PRINT "TIEMPO ORDENACION PARA ";N;" ELEM
ENTOS : ";MID$(T$,3,2);":";MID$(T$,5,2)
200 OPEN 4,4:PRINT#4,CHR$(14)
210 PRINT#4,"TABLA INICIAL  CONTADORES  TABL
A FINAL"
220 PRINT#4,"--------------------------------
-------"
230 FORI=1TON
240 PRINT#4,T1(I),T2(I),T3(I)
250 NEXT
260 PRINT#4
270 PRINT#4,"TIEMPO ORDENACION PARA ";N;" EL
EMENTOS : ";MID$(T$,3,2);":";MID$(T$,5,2)
READY.
```

dentro de los métodos de clasificación. Si bien, en un principio, todos los métodos parten de ideas sumamente sencillas que dan rendimientos pobres, son susceptibles de mejoras que pueden incrementar notablemente su rendimiento. Esto no sucede con la familia de clasificación que estamos considerando pero, teniendo en cuenta su sencillez, podemos dedicarle algo de espacio.

Como os he dicho anteriormente, la idea del método es que si un elemento es mayor que otros trece quiere decir que ha de ocupar el puesto 14 en la tabla ordenada. Así pues, si *comparamos cada par de elementos* y contamos cuántos son menores que cada uno en concreto, tendremos un método de ordenación en donde el contador nos dará, al final, la situación relativa que tiene que ocupar cada uno de los elementos. El algoritmo que desarrollaremos deberá tener en cuenta la posibilidad de que haya elementos iguales, lo cual no es demasiado difícil teniendo en cuenta que si cada par de elementos sólo se compara una vez y en cada comparación se incrementa uno u otro contador, según un mismo criterio y siguiendo siempre el mismo orden, *llegaremos a un resultado final* con todos los contadores distintos aunque haya elementos repetidos. En la figura aparece el listado de una rutina de clasificación según este método.

De la observación de la rutina se deduce cuál es la principal ventaja del método, que es su tremenda sencillez. Por contra, su principal (y definitiva) desventaja es su bajísimo rendimiento a partir de valores de **N** (número 'de elementos de la tabla) relativamente grandes, como podréis observar en la tabla resumen que publicaremos con el último de los métodos de esta superfamilia.

El bajo rendimiento se debe a que cada elemento se ha de comparar con todos los otros, lo cual quiere decir que si el tamaño de la lista es **N** se han de hacer **N**(**N**-1)/2 comparaciones, lo cual nos da unos tiempos de ejecución proporcionales a **N** ^ 2 (al doblar el tamaño de la lista se multiplica por cuatro el tiempo de ejecución). El propio sistema seguido en el algoritmo nos indica que el método es completamente insensible a la distribución inicial de los elementos de la lista, es decir, su rendimiento sólo depende del tamaño de la lista, y es independiente de la ordenación inicial.

Tabla inicial	Contadores	Tabla final
53	9	2
81	13	8
79	12	11
2	0	16
92	18	20
54	10	29
16	3	42
42	6	44
11	2	49
49	8	53
98	19	54
8	1	68
87	16	79
29	5	81
44	7	81
20	4	86
81	14	87
87	17	87
86	15	92
68	11	98

Tiempo ordenación para 20 elementos: 0:03.

Tabla inicial	Contadores	Tabla final
10	2	2
8	1	8
46	16	10
27	10	11
41	14	11
62	20	19
24	8	23
50	18	23
19	5	24
98	27	26
89	25	27
99	29	30
2	0	31
38	13	38
93	26	41
11	3	41
31	12	46
54	19	46
11	4	50
79	22	54
30	11	62
46	17	76
83	23	79
41	15	83
98	28	84
84	24	89
76	21	93
26	9	98
23	6	98
23	7	99

Tiempo ordenación para 30 elementos: 0:07.

(termina en la pág. siguiente)

MARKETCLUB

● Vendo aplicación de facturación con control de representantes, 9 listados, 6 ficheros, estadísticas, etc., permite copias de seguridad. Configuración: VIC, 8K, disco e impresora, 40.000 Ptas. Escribir a Jaime Ameller Pons. General Mola, 15, 1º B. CALATAYUD (Zaragoza).

● Vendo interfaz y programa para RTTY y CW para el PET a 15 K. Rafael, EA3CGK - Avda. Barcelona, 21, A, 4º 2ª. IGUALADA (Barcelona).

● Se busca experto en VIC-20 para colaborar en la creación y coordinación de un Club de Usuarios de VIC en Barcelona. Llamar a Srta. Rosa Romero. Teléfono 211 54 40.

● Vendo cartucho 16K VIC-20, por 14.000 ptas. Hago programas en Basic de Commodore (todas las versiones) bajo encargo. Desearía contactar con usuarios de Commodore en la zona de Madrid, para cambio de programas, impresiones, pokes especiales, etc... Razón: Francisco Gutiérrez. Santiago Rusiñol, 12. MADRID-3. Teléf. (91) 253 13 40. Horas comida y cena.

¡SOCORRO!

el esquema del VIC-20

Como se puede observar, publicamos los esquemas del VIC por entregas para que puedan ser lo más legibles que permite el formato de nuestra Revista. No obstante, en la versión original forman un todo (y dentro del VIC también, como es lógico). Para que se pueda seguir con facilidad el recorrido de una determinada señal, los conductores que «salen» de cada una de las tres hojas en que está dividido el esquema, terminan en una indicación que nos dice a qué otro (u otros) circuitos va conectada esta señal. En esta edición se publican esquemas en ambas contraportadas interiores.

técnicas de ordenación... (conclusión)

Tabla inicial	Contadores	Tabla final
31	10	0
83	32	0
80	30	6
54	25	13
23	7	16
37	12	18
99	37	19
74	28	23
71	27	29
81	31	30
37	13	31
92	35	34
95	36	37
99	38	37
34	11	39
16	4	40
43	18	42
50	24	42
13	3	43
99	39	44
89	33	46
44	19	47
0	0	48
42	16	49
56	26	50
18	5	54
19	6	56
40	15	71
47	21	74
89	34	79
79	29	80
49	23	81
0	1	83
30	9	89
46	20	89
48	22	92
6	2	95
39	14	99
29	8	99
42	17	99

Tiempo ordenación para 40 elementos: 0:13.

Tabla inicial	Contadores	Tabla final
29	12	0
32	14	1
1	1	3
62	30	6
83	38	7
93	46	8
83	39	10
6	3	17
3	2	18
92	45	23
96	49	23
58	28	26
7	4	29
45	22	29
0	0	32
57	27	32
26	11	34
54	25	37
54	26	41
23	9	41
44	21	41
93	47	44
41	18	45
10	6	45
76	35	52
32	15	54

(pasa a la siguiente)

Tabla inicial	Contadores	Tabla final
87	44	54
78	36	57
74	33	58
23	10	61
17	7	62
52	24	63
69	32	69
61	29	74
18	8	74
41	19	76
37	17	78
94	48	79
86	43	83
79	37	83
29	13	83
45	23	84
83	40	85
34	16	86
84	41	87
63	31	92
8	5	93
41	20	93
85	42	94
74	34	96

Tiempo ordenación para 50 elementos: 0:21. ■

NOTICIAS

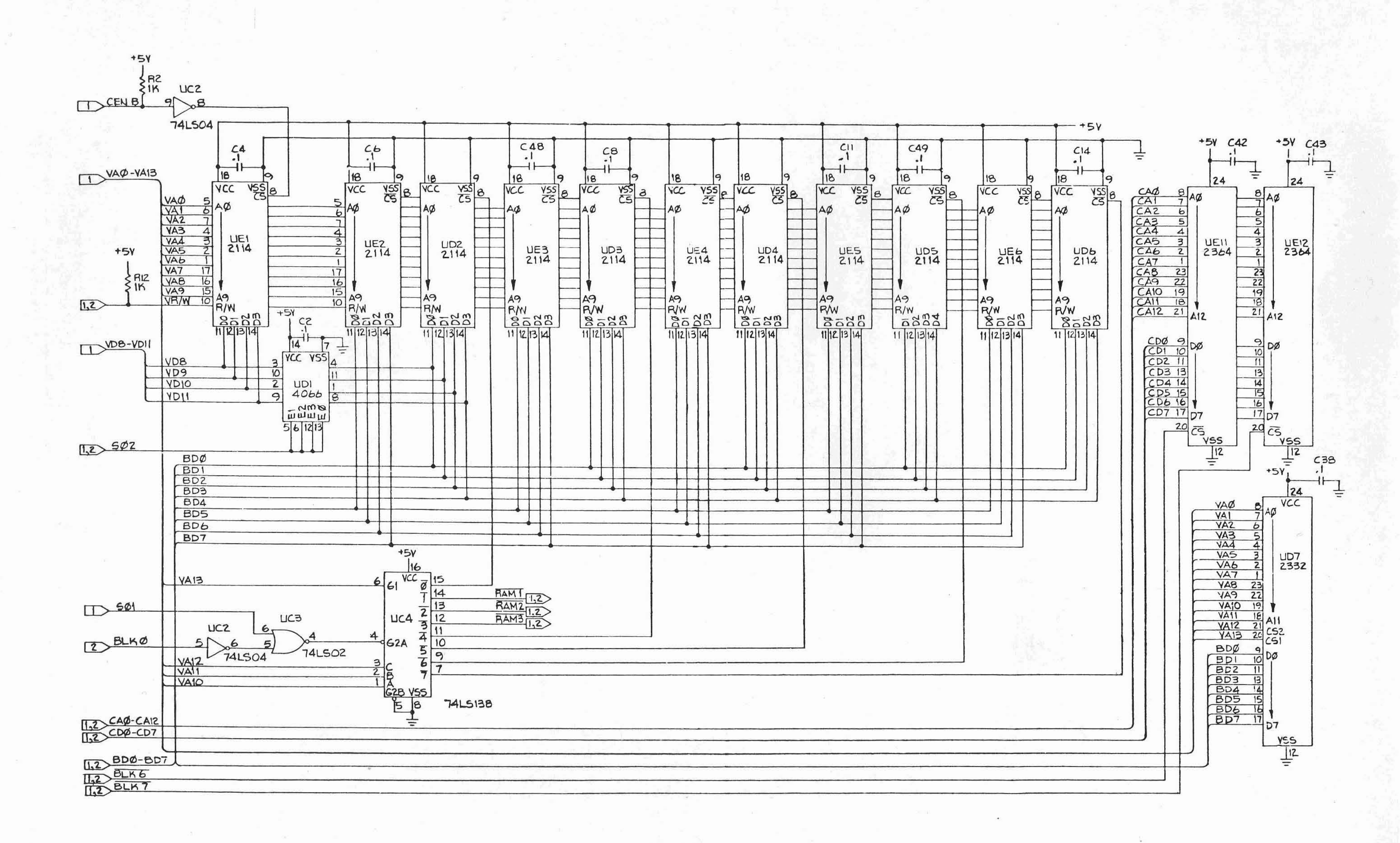
los números atrasados de CLUB COMMODORE

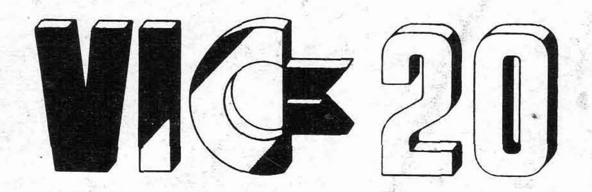
Dada la apabullante demanda de números atrasados de nuestra Revista, está en estudio la edición en fotocopias de los números publicados en 1982 (el 0, el 1, el 2 y el 3) para evitar la especulación y el mercado negro. Como somos muy previsores hemos reservado un ejemplar de cada número para esta edición. Para información de los curiosos diremos que ahora sólo disponemos de los números 4 y 5 (y naturalmente el 6). Los demás están TODOS agotados.

ESQUEMA DEL VIC-20

Circuit 3

Circuit 3





PROGRAMAS VIC-20

Ref.	Programa	P.V.P. Ptas.
	Libro curso de introducción al Basic Parte I Con dos cintas o un disco.	2.500
	Libro curso de introducción al Basic Parte II Con dos cintas o un disco.	2.500
D-1001	**Agenda** Necesita ampliación de 8K.	5.000
D-1002	**QSL** Necesita ampliación de 8K.	3.000
D-1003	**Test Demo**	3.000
D-1004	**Assembler** Necesita ampliación de 3 K.	5.000

CINTAS PARA CASSETTE

Ref.	Programa	P.V.P. Ptas.
C-125	**Hangmath** No necesita ampliación.	1.500
C-128	**Programación lineal** No necesita ampliación.	500
C-129	**Matrices** No necesita ampliación.	500
C-130	**Caja** Necesita ampliación de 16K	1.500
C-131	**Regresiones I** No necesita ampliación.	500
C-132	**Regresiones II** No necesita ampliación.	500
C-133	**Estadística I** No necesita ampliación.	500
C-134	**Estadística II** No necesita ampliación.	500
C-135	**Sistemas** No necesita ampliación.	500
C-136	**Dieta** Necesita ampliación de 8K.	1.500
C-137	**Integración** No necesita ampliación.	500
C-139	**Vicalc** No necesita ampliación.	1.500
C-140	**Skymath** Necesita ampliación de 3K.	1.500
C-141	**Space Division** Necesita ampliación de 3K.	1.500
C-142A	**Interface y programa CW** No necesita ampliación.	25.000
C-142B	**Programa RTTY** No necesita ampliación.	2.500
C-143	**English Language** Necesita ampliación de 8K.	2.000
C-144	**Quiz Master** Necesita ampliación de 8K.	2.000
C-145	**Mastermind** Necesita ampliación de 8K.	3.000
	Además para esta cassette se están preparando los siguientes temas:	
C-145A	**Gastronomía**	1.500
C-145B	**Música**	1.500
C-145C	**Cinematografía**	1.500
C-145D	**Deportes** Etc...	1.500
C-201	**Codemaker** No necesita ampliación.	1.500
C-202	**Wall Street** No necesita ampliación.	1.500
C-203	**Simple Simon** No necesita ampliación.	1.500
C-204	**Damas** No necesita ampliación.	1.500
C-205	**Alien Blitz** No necesita ampliación.	1.500
C-206	**Kosmic Kamikaze** Necesita ampliación de 3 u 8K, dado que hay dos versiones del juego.	1.500
C-207	**Star Wars** No necesita ampliación de memoria.	1.500
C-208	**Amok** No necesita ampliación.	1.500
C-209	**The Alien** Necesita ampliación de 3K.	1.500
C-210	**Invader Fall** No necesita ampliación.	1.500
C-211	**A-MAZ-ING** Necesita ampliación de 3K.	1.500
C-212	**Math-Hurdler, Monster Maze** No necesita ampliación.	1.500
C-213	**Golf** Necesita ampliación de 3K.	1.500
C-215	**VIC GAMES II** No necesita ampliación.	1.500

JUEGOS EN CARTUCHO

Ref.	Programa	P.V.P. Ptas.
1904	**Super Slot** Se juega por teclado o con Joystick.	4.500
1906	**Super Allien** Se juega por teclado o con Joystick.	4.500
1907	**Júpiter Lander** Se juega por teclado.	4.500
1908	**Draw Poker** Se juega por teclado.	4.500
1901	**Avenger** Se juega por teclado o con Joystick.	4.500
1909	**Road race** Se juega por teclado.	4.500

PROGRAMAS PROFESIONALES

Ref.	Programa	P.V.P. Ptas.
Z0001	**Contabilidad S/V-20** **Stock-VIC**	
V0501	**Control películas para Video-Clubs**	
Z0601	**VIC/Entrapunt**	

Nota: Para más detalles sobre estos programas consultar.

NOVEDADES ENERO 1983

PROGRAMAS EN CINTA:

Ref.	Programa	P.V.P. Ptas.
C-146	**Matemáticas 1** (nivel BUP) Necesita ampliación de 8 ó 16K.	2.000
C-216	**Escape** No necesita ampliación.	1.500
C-217	**Biocomp** Necesita ampliación de 3K.	2.000
C-218	**Cubo VIC** No necesita ampliación.	1.500
C-219	**Type a tune** No necesita ampliación.	2.000

PROGRAMAS EN CARTUCHO:

Ref.	Programa	P.V.P. Ptas.
1911	**The sky is failling** Es necesario utilizar Paddle.	4.500
1902	**Star battle** Se juega con teclado o con Joystick.	4.500
1919	**Sargon II chess** Se juega con teclado o con Joystick.	4.500
1924	**Omega race** Se juega con Joystick.	4.500
C-400	**VIC FORTH**	consultar
C-401	**VIC STAT**	consultar
C-402	**VIC GRAPH**	consultar
1912	**Mole attack** Sólo para teclado.	4.500
1909	**Radar ratrace** Se juega con teclado o con Joystick.	4.500
C-403	**WORDCRAFT-VIC** (compatible series CBM 8000 y CBM 4000)	42.500
C-404	**VIC SCREEN MASTER**	10.000

PROGRAMAS EN DISCO

Ref.	Programa	P.V.P. Ptas.
D-1005	**English Language** Necesita ampliación de 8K.	2.500
D-1006	**Quiz master** Necesita ampliación de 8K.	2.500
D-1007	**Matemáticas I** (nivel de BUP) Necesita ampliación de 8K.	2.500
D-2000	**VIC WRITER** Necesita ampliación de 8K.	13.000
D-2001	**SIMPLICALC** Necesita ampliación de 8K.	13.000
D-2002	**VIC FILE** Necesita ampliación de 8K.	13.000

Depósito Legal Sep. B. 2133-1958 - Altés, s.l., Caballero 87, Barcelona